MULTI-AXIS NUMERICAL CONTROL

A thesis submitted for the degree of

Master of Engineering

in the

University of Canterbury

New Zealand

by Ma Li

1989

ACKNOWLEDGEMENTS

I am indebted to my supervisor Dr G R Dunlop for his guidance, encouragement and help during the course of this research and preparation for this thesis. I am grateful to Mr N Bolland for his sincere concern and help.

I wish especially to present this to my husband Xiao Xuan, for his strong encouragement and support, and to all of my family for their significant help.

Finally, my thanks go to all those who assisted me in many ways with their knowledge, time and enthusiasm.

CONTENTS

ACKNOWLEDGEMENTS I
ABSTRACT IV
PUBLISHED PAPERS V
LIST OF SYMBOLS VI

CHAPTER 1 INTRODUCTION 1
1.1 Robotics 2
1.2 Satellite Tracking – The "Key Hole" Problem 3
1.3 The Stewart Platform 5
CHAPTER 2 NUMERICAL CONTROL OF MULTI-AXIS MACHINERY 13
2.1 Introduction 13
2.2 Single Motor Control Problem 14
2.3 Multiple Motor Control Problem 17
2.4 Accurate Multiple Motor Control 18
CHAPTER 3 DESIGN OF A MULTI-MOTOR CONTROLLER 26
3.1 Introduction 26
3.2 The Computer and Its I/O Channel 27
3.3 The HCTL-1000 28
3.3.1 The Operation 28
3.3.2 User Accessible Registers 29
3.3.3 Timing Diagram 30
3.4 Design of the Controller 31
3.4.1 IBM-PC I/O Bus 31
3.4.2 Decoder and Control Logic 31
3.4.2.1 Clock 31
3.4.2.2 Control Signals Decoding 32
3.4.2.3 Data Transfer 33
3.4.2.4 An Example 33
3.4.3 Reset 33
3.4.4 HCTL-1000 34
3.4.5 Buffer 35
3.4.6 Controller Socket 35
3.5 Testing 35
CHAPTER 4 DC MOTOR PWM SERVO DRIVE 51
4.1 Introduction 51

4.2 DC Motor 52
4.2.1 Operation Principles 52
4.2.2 Classification and Characteristics 53
4.2.2.1 Shunt Motor 54
4.2.2.2 Series Motor 54
4.2.2.3 Compound Motor 55
4.2.2.4 Separately Excited Motor 55
4.2.2.5 Constant Field and Permanent Magnetic DC Motors 55
4.2.2.6 Motor Parameters 57
4.3 DC Motor PWM Servo Drive 58
4.3.1 Introduction 58
4.3.2 Three Modes of PWM Drives 59
4.4 Design and Testing 59
4.4.1 Bipolar PWM Servo Drive 59
4.4.1.1 Logic Design 59
4.4.1.2 PWM Drive Design 61
4.4.1.3 Problem and Analysis 62
4.4.2 Limited Unipolar PWM Servo Drive 62
4.4.3 Motor Performance 63
4.5 Summary 64
CHAPTER 5 SYSTEM MODELLING AND ANALYSIS 82
5.1 Control Modes 82
5.1.1 Position Control 82
5.1.2 Proportional Velocity Control 83
5.1.3 Integral Velocity Control 84
5.1.4 Trapezoidal Profile Control 84
5.2 The Electromechanical Model 85
5.2.1 The Motor 85
5.2.2 The Encoder 87
5.2.3 The Control System Modelling 88
5.2.4 Theoretical Response 91
5.3 Experimental Response Analysis 92
5.4 Summary 93
CHAPTER 6 CONCLUSION 105
APPENDIX A SYSTEM PERFORMANCE FOR LARGE STEP INPUTS 107
APPENDIX B ANALYSIS OF FRICTION EFFECTS 109
APPENDIX C PROGRAM LISTING 113
REFERENCES 123

ABSTRACT

This thesis presents the analysis, design and development of a multi-axis machinery numerical control system. The purpose of this research is to provide a numerical control method to overcome the multi-axis numerical control problem.

The control system includes an IBM-PC computer as a host processor, a plug in multi-motor controller board based on commercial numerical motor controller ICs. These implement a software specified digital control algorithm and output a PWM control number. Six motors, their driven actuators, and digital incremental feedback encoders complete the system.

The experimental aspects of the work included the design of the IBM-PC plug in motor controller board and the motor drive board, computer software for real-time control and the system testing. For the theoretical aspects of the work, control theory was used to develop the mathematical model of the system. This aimed at providing a tool to predict and optimize the system performance, therefore, to fulfill the high positioning and high precision control task.

PUBLISHED PAPERS

Dunlop G R and Ma Li: Multi-axis Numerical Control. Proc. IMC Conf., Christchurch, New Zealand, May 1988

Afzulpurkar N V, Ma Li, Dunlop G R and Johnson G R: Design of Parallel Link Robot. Proc. NELCON Conf., Christchurch, New Zealand, August 1988

LIST OF SYMBOLS

Az	$90^\circ - \phi$, azimuth
El	$90^\circ - \theta$, elevation
T(f)	(step motor) pull-out torque at a stepping rate f
$T_1(f)$	load torque
J	motor inertia
$\theta(t)$	rotating angle of motor shaft
d	microprocessor direct control delay value
T_1	time taken in the idle path (microprocessor direct control)
T_2	time taken in the step path (microprocessor direct control)
ω_1	motor velocity in one motor direct control
ω_6	motor velocity in one motor direct control
T_m	motor shaft torque
Φ	DC motor magnetic flux per pole
I_a	motor armature current
K_p	constant of the motor winding
E	motor back emf voltage
ω	motor shaft velocity
R_a	motor armature resistance
V_m	motor terminal voltage
K_t	motor torque constant
K_e	motor back emf voltage constant
P_m	mechanically developed motor power
P_e	electrical power absorbed in the motor rotor
Vs	the voltage of the power supply
I_{AB}	the current flowing through the motor in PWM servo drive
I_{on}	the current flowing through the motor during the FETs "on" time in PWM servo drive
I_{off}	the current flowing through the motor during the FETs "off" time in PWM servo drive
V_{in}	PWM servo drive input voltage
t	time
$\frac{d\omega}{dt}$	motor shaft angular acceleration
K_m	the motor constant

τ	the motor time constant
K	the gain number in the register R22H in the HCTL-1000
A	the zero number in the register R20H in the HCTL-1000
B	the pole number in the register R21H in the HCTL-1000
T	sample time
A_p	amplitude of a step input
T_{fr}	torque caused by friction
V_{ef}	the effective voltage
α	the effective voltage coefficient
K_{ex}	the slope of the experimental response
K_{th}	the slope of the theoretical response
PWM%	the PWM duty cycle

CHAPTER 1

INTRODUCTION

Despite rapid developments in electronics and communications, traditional mechanical engineering is still an essential part of modern society. In particular, Multi-axis machinery is an important field which draws on all of these engineering disciplines.

Multi-axis machinery is a type of machinery which has multiple axes or mechanisms, each of which can execute a required motion to produce a coordinated motion for the whole machine. The purpose of this type of machine is to do complicated jobs so as to increase efficiency or reduce costs. Robots, satellite tracking mechanisms and some machine tools are typical applications of such machines.

The remaining part of this chapter will discuss these applications including their functions and existing problems, and then give a mechanism as a solution. The second chapter discusses the numerical control problems associated with ordinary control means used to drive multi-axis machinery. A satisfactory solution to these problems is developed.

The later chapters present the design, theoretical analysis and performance of a multi-axis machine numerical controller developed during this research, i.e. the problems raised in CHAPTERs 1 and 2 will be solved.

1.1 ROBOTICS

Since the word "robot" was first introduced in 1917 (McCloy 1986), it has become widely used by many people. Because of the potential shown by robots in substituting people's labour especially in dangerous or boring work, the use of robots is becoming widespread.

According to mechanical arrangement of the manipulators, robots can be divided into two categories: i) The serial or open link manipulators and ii) the parallel or closed link manipulators.

The arrangement of a six degrees of freedom serial link robot is shown in Fig. 1.1. It has six fixed length links and is a traditional type coming from the simulation of the human body, i.e. anthropomorphic. Each link can swing through an arc with respect to the preceding link. Hence, the manipulator has six degrees of freedom and the end effector (or gripper) is positioned within the working volume.

The main advantages of a serial link manipulator are:

-the manipulator has high mobility, i.e., has a long reach and a large range of motion since the links are connected one after another;

-it has flexible reaching points, even to corners.

The disadvantages are:

-the cantilever construction subjects the manipulator to large bending moments, which results in link deflections at full reach. This decreases the accuracy of positioning and limit the application under heavy loads;

-a given set of end actuator coordinates does not have a unique set of joint coordinates. In other words, the link positions are usually indeterminate (c.f. Fig. 1.2) (Dunlop & Afzulpurkar 1988).

1.2 SATELLITE TRACKING – THE "KEY HOLE" PROBLEM

Satellites are now the dominant carriers of telecommunications and are playing important parts in space and the earth resource exploitation. Satellite tracking systems have been developed since 1963 after the space research started in late fifties. Good tracking systems must be able to find their satellites quickly and maintain communication with them through unexpected orbital changes. For maritime earth stations, they need also be able to overcome the effect of rolling and pitching of the ships. The earth station antenna dish diameters may be as large as 30m and can weigh up to several hundred tons (Afzulpurkar et al 1988). As the frequency used increases, the beamwidth is reduced, so more than a fraction of a degree of apparent satellite movement must be tracked (Pratt and Bostian 1986). Hence a good tracking mechanism is required to ensure the completion of the task. On the other hand, they must be relatively inexpensive.

The present popularly used tracking mechanisms can be generally divided into three categories: Alt-Azimuth or Az-El mount (c.f. Fig. 1.3), X-Y mount (c.f. Fig. 1.4) and a combination of these two (c.f. Fig. 1.5 and Miya 1981).

The path followed by any satellite moving around the earth is a decaying ellipse in the orbital plane. The coordinates to which the antenna must be pointed to communicate with the satellite are called the "look angles". These are most commonly specified as "azimuth" (Az) and "elevation" (El). Azimuth ($90^{\circ} - \phi$) is measured eastward from

geographic north to the intersection of the satellite orbital plane on the horizontal plane at the earth station. Elevation ($90^{\circ} - \theta$) is the angle measured above the horizontal plane to the line of sight of the satellite (c.f. Fig. 1.6).

An Az-El type of mounting system has two axes, Az and El. The antenna bore sight axis will trace a semicircle on a vertical plane about El axis and can sweep a horizontal circle about Az axis. When a satellite starts approaching to the zenith, i.e. the El angle 90° (c.f. Fig. 1.7(a)), the antenna dish has to rotate the azimuth angle through 180° about the bore sight axis to continue the tracking because of limitations in the mechanism (c.f. Fig. 1.7(b)) . Thus the system has a singularity at the zenith. During the time taken for this 180° azimuth rotation, the satellite moves out of the beam of a high gain antenna and the station loses contact with the satellite. This is known as the "key hole" problem. The problem becomes particularly severe when the tracking system is mounted on a ship. The rolling and pitching action of a ship causes the singularity of the Az-El mount to trace out a flattened cone about the zenith. If the satellite is within this flattened cone region, communication contact can be lost and the tracking ceases. When receiving signals from deep space probes, there is only one chance of capturing the data sent by the space craft and such a loss of contact is unacceptable (Afzulpurkar et al 1988).

An X-Y type of mounting system has X and Y axes (c.f. Fig. 1.4). When the antenna dish tracks a satellite along the horizontal plane and near the axis mountings, the trackin path is occluded by the mountings, so it has to rotate 180° for continuing the tracking. Thus the "key hole" problem exists along the horizontal axis in an X-Y mounting system. For overcoming the effect of rolling and pitching of a ship mounted application, the antenna being able to point below the horizon is needed, so the X-Y mounting is incapable of fulfilling this kind of tracking requirements. To solve the

"key hole" problem for deep space work, two ground based X-Y mounted antennas are mounted perpendicular to each other. Thus the singularity of each antenna is covered by the other and the signals are received and complementary mutually. However, the cost of the tracking system is doubled.

The conventional solution to the "key hole" problem is to use a combined mount of X-Y type and Az-El type (c.f. Fig. 1.5(c) and Miya). A horizontal plane is always maintained against ship's motions by controlling the X and Y axes of the plane. The direction to the satellite is produced by controlling the azimuth-elevation axes mounted on this horizontal plane. Controlling of the X-Y and Az-El axes is independent, but the cost and complexity of the operation are greatly increased.

1.3 THE STEWART PLATFORM

The solution to the above problems in robotics and satellite tracking systems is to use a Stewart platform mechanism (c.f. Fig. 1.8).

The Stewart platform was proposed in 1965 (Stewart) and is widely used for aircraft simulation and other purposes nowadays. It is a parallel link manipulator. It consists of two bodies connected by six legs. One body is called the base, which is usually fixed, and the other is called the platform which is movable. Each of the legs is in the form of a linear actuator. Each actuator has one end joined to the base and the other end joined to the platform. The actuators are prismatic joints and can expand and contract independently thus positioning the platform in space. This arrangement provides a sturdy mechanism with six degrees of freedom which means it can take any

position and any orientation within a certain work volume. The arrangement of the structure is symmetric to achieve maximum stiffness.

With this rigid mechanism, the joint deflection problem in serial link robotics is considerably reduced and high accuracy positioning can be achieved. These properties make it suitable for use in heavy duty environments, such as automotive assembly applications and the like (c.f. Fig. 1.9).

For the satellite tracking application, an antenna dish is mounted on the platform and a tracking system with six degrees of freedom is formed. The antenna dish can even point 30° below the horizon with carefully designed joints (c.f. Afzulpurkar et al 1988). When this mechanism is applied in a ship mounted environment, it is able to track the satellite rapidly without mechanism singularity limitations, therefore is a rather simple and reliable mechanism to overcome the "key hole" problem (c.f. Fig. 1.10). The six identical actuators simplify the manufacture.

So far the solution to the problem raised in 1.1 and 1.2 has been clearly given – the Stewart platform.

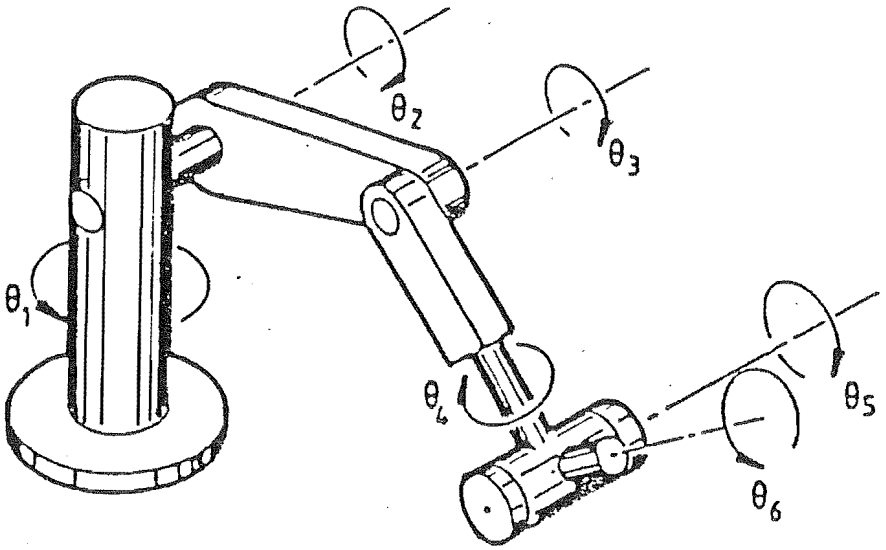


Fig. 1.1 Typical seriel link manipulator
with six degree of freedom

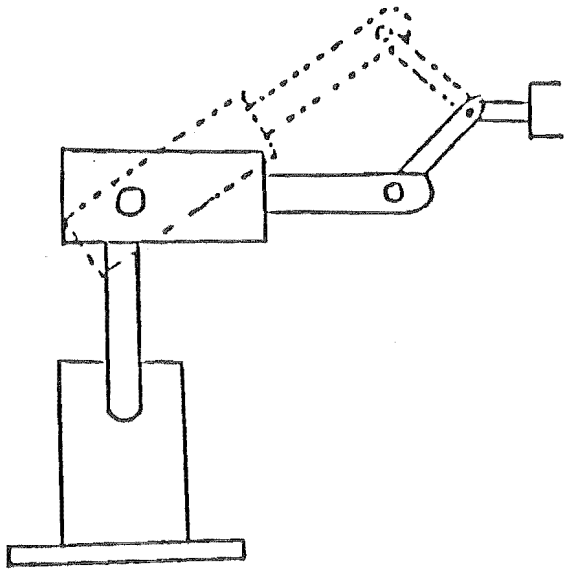


Fig. 1.2 Kinematic independency in
open link manipulator

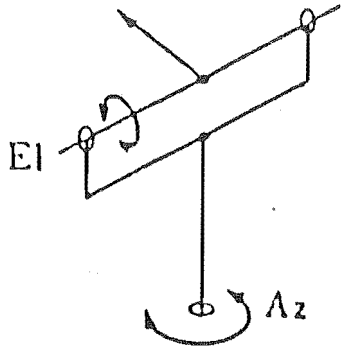


Fig. 1.3 Az-El antenna mount system. 2 axes

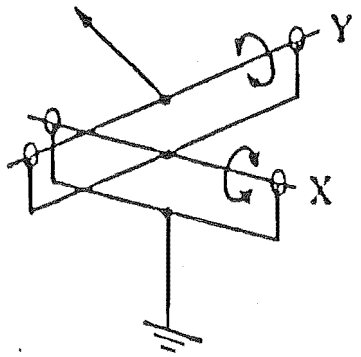


Fig. 1.4 X-Y antenna mount system. 2 axes

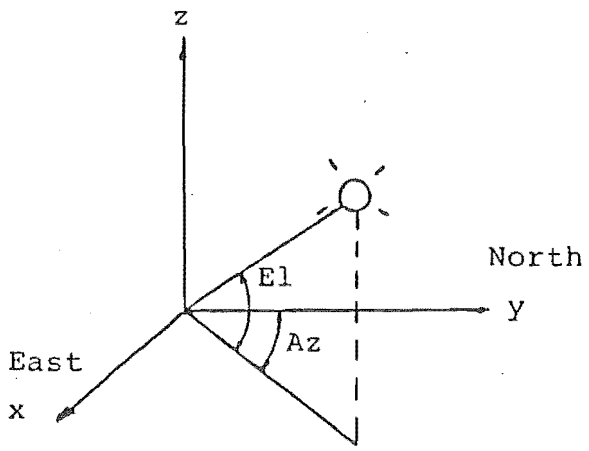
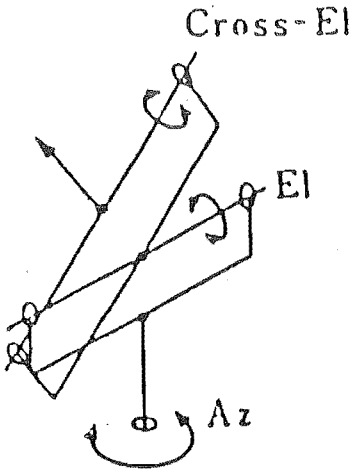
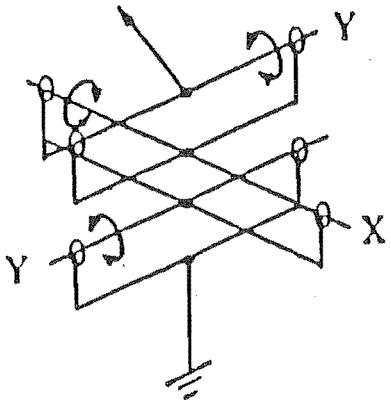


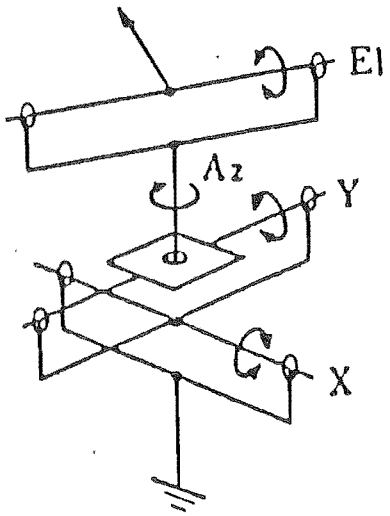
Fig. 1.6 Look angles



(a) Az, El, Cross-El mount. 3 axes



(b) X-Y-Y' mount
3 axes



(c) X-Y and Az-El
4 axes

Fig. 1.5 Antenna mixed mount systems

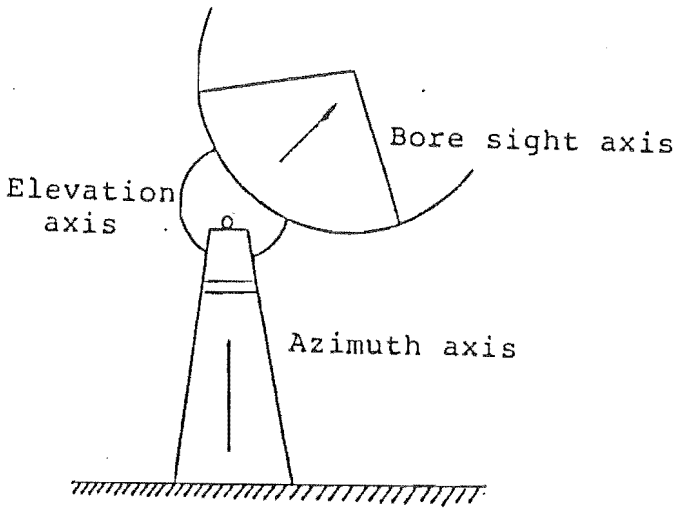


Fig. 1.7(a) Alt-Azimuth antenna mount

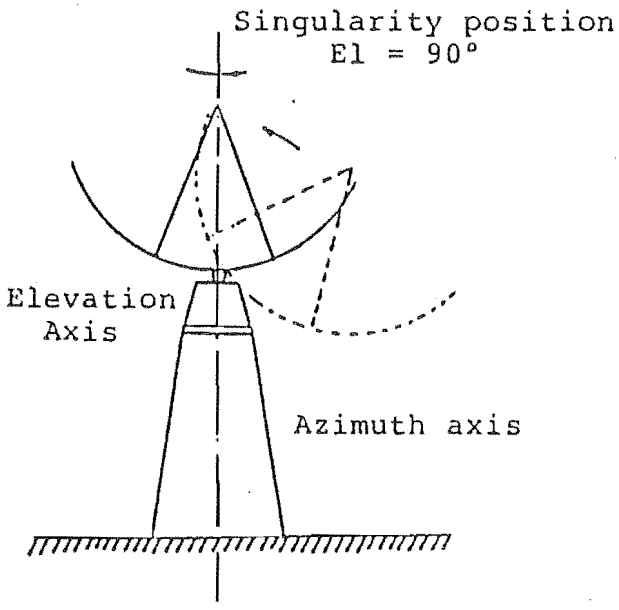


Fig. 1.7(b) Singularity position of Alt-Azimuth antenna mount

Fig. 1.8 The Stewart platform
in the project

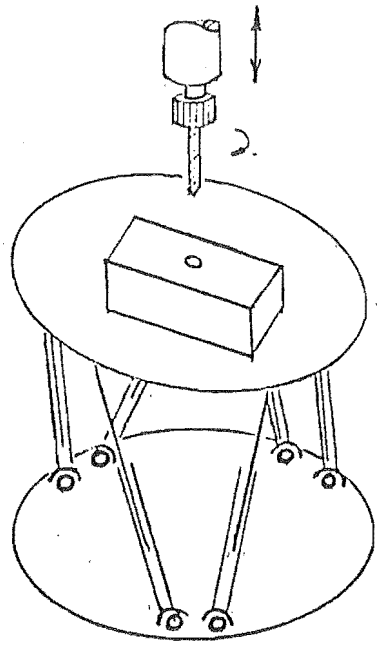


Fig. 1.9 Parallel robotic based on Stewart platform

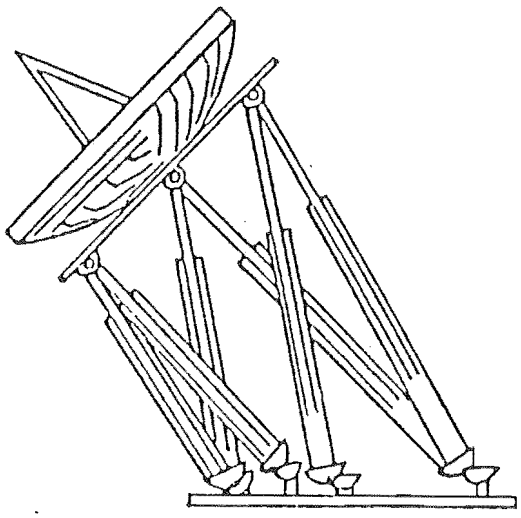


Fig. 1.10 Satellite tracking system based on Stewart platform

CHAPTER 2

NUMERICAL CONTROL OF MULTI-AXIS MACHINERY

2.1 INTRODUCTION

Some method is required for controlling and positioning the Stewart platform or any other multi-axis machinery. The six actuators of the Stewart platform are driven each by a DC motor. The numerical control system discussed here is not confined only to DC motors, but can be extended to other driving components like DC brushless motors, step motors and hydraulic cylinders. A general discussion about the numerical control to multiple axis machinery in a broader sense will be given.

Hydraulic cylinders and DC motors can use the same type of control signal i.e. a voltage or a PWM signal. Step motors require precise phase switching information and some form of current or voltage control. Brushless DC motor can be treated as a class of step motor.

To control these motors or hydraulic cylinders, a microprocessor is a good choice because of its low cost and high reliabilities. However, the relatively slow speed of most microprocessors results in limited control for more than one motor/actuator. The various features are analyzed in following sections.

2.2 SINGLE MOTOR CONTROL PROBLEM

Take a step motor directly controlled by a microprocessor software timing as an example, where several features must be considered. The typical pull-out torque/speed characteristics of a step motor are shown in Fig. 2.1. This represents the maximum torque which the motor can develop at each operating speed. If the load torque exceeds the pull-out torque, the rotor is pulled out of synchronism with the magnetic field and the motor stalls or even reverses. However, for velocity profiling operations, timing is critical, more factors have to be considered. Fig. 2.2 shows the relationship between pull-out torque and load torque. It can also be algebraically expressed as follows (Acarnley 1982):

$$T(f) = T_1(f) + J\frac{d^2\theta}{dt^2} \qquad (2.1)$$

Where, T(f) – the pull-out torque at a stepping rate f;

$T_1(f)$ – the load torque;

J – the whole system inertia;

$\theta(t)$ – the rotating angle of the motor.

That means, to move the load and change its velocity at the required rate, the being developed torque of the motor must be large enough to overcome the load torque and accelerates the system inertia, thus to keep the consistent accuracy and stability, which are the essential requirements in step motor control. Otherwise, the motor either fails to operate at all or drops out of step during acceleration and deceleration, causing permanent errors (Acarnley 1982; Dunlop 1986).

To perform each specified movement during a microprocessor software direct control, the step rate (steps/sec) and the target distance in steps are calculated. The latter one

can be calculated by knowing the step length. The former one can be obtained from the following method based on the one of Acarnley (1982).

The flowchart shown in Fig. 2.3 is the method giving the step interval by software for direct control of one motor. At regular step intervals, the microprocessor software produces a phase on/off pattern and sends it to the step motor phase switches. The step interval can be calculated from the following:

$$\text{Step Interval} = dT_1 + T_2 \tag{2.2}$$

Where, T_1 – the time taken to go through the idle path;

T_2 – the time taken to go through the step path;

d – delay value.

The step interval delay value d is placed in a loop counter initially, then the step interval is established by decrementing the counter and producing a new phase on/off pattern each time the loop counter reaches zero. Note that the loop takes typically 10μs per loop cycle going through the idle path except on the last cycle which takes a typical extra 50μs going through the step path. The addition of "DelayS" is to keep the time taken in step path a constant no matter whether or not a new table is set up. The "DelayL" here can be zero and must be increased in multiple motor control which is discussed in the next section. Then the stepping rate can be calculated by the following:

$$\text{Step Rate (steps/sec)} = \frac{1}{\text{Step Interval (sec/step)}} \tag{2.3}$$

The timing quantization is introduced by this microprocessor software loop. The effect on performance is illustrated by the calculated results given in Table 2.1 (Dunlop and Ma 1988). Note that:

1) The maximum speed a motor can achieve may be limited by the operating speed of the microprocessor to less than the one determined by the load.

2) At high step rates, the velocity changes (or acceleration) which must occur within one step time (so as to maintain synchronism) are very large. For every 10 times increase of the speed, the required acceleration increases 1,000 times. In other words, a large torque is required to achieve large speed change when the speed is high; while a small torque is required to achieve small speed change when the speed is low.

Nevertheless, this is contrary to the pull-out torque/speed characteristic of a step motor. That is, the torque T(f) available from the motor is greatly reduced when the acceleration required to achieve the speed change is largest at higher speeds (c.f. Fig. 2.2). At slow speeds, the acceleration is much less and the torque available is much greater. A software based microprocessor system can give satisfactory control of the velocity profile up to stepping rates of 10,000 steps/sec, but may limit high-speed performance (Acarnley 1982).

The use of a faster microprocessor to reduce the loop and calculation times will increase the upper speed limit and reduce the differences between velocity changes within one step time. However, in the final analyses, the inherent timing quantization rather than the potential motor performance limits the control of the motor speed predominantly.

3) The speed resolution decreases as the speed increases. For example, at 100 steps/s, a unity change in step interval produces a 0.1% change in stepping rate, but at 1,000 steps/s, it gives a 1% difference in stepping rate. The accuracy of speed control is decreased when the speed is high.

2.3 MULTIPLE MOTOR CONTROL PROBLEM

The method of direct control of more than one motor by a single microprocessor is shown in Fig. 2.4 (Dunlop and Ma). The DelayL shown in Fig. 2.3 must be increased so that the time taken in any motor control block is a constant irrespective of whether the step output or idle loop path is taken through the flow chart. If this is not done, then the 50μs overhead for the first motor's step generation can occur anywhere within the timing loop for the second motor (Dunlop 1986). This causes a variation in the stepping rate of the second motor, which in turn affects the stepping rate of the first motor. The effect of introducing a 60 μs (50+10) loop delay for each motor is calculated in Table 2.2. The figures are for a typical microprocessor controlling six motors. Compare with Table 2.1 and it will be seen that the three problems associated with direct motor control of one motor are greatly multiplied for six motors:

1) The maximum velocities attainable are reduced, for the example in Fig.2.3, the velocity drops from 16,667 to 2,778 steps/s.

2) High acceleration is required at lower speeds. The upper speed limit given in Table 2.2 for six controlled motors is 2778 steps/sec. The same acceleration required (1.9E6) to reach this speed would produce a single motor speed of about 5600 steps/sec, or more than double the six motor control speed limit. Allowing for an approximate halving of the torque for a doubling of the speed, then a torque limit of 1E6 yields a single motor speed of 4545 steps/sec or a 60% increase in the speed possible with six motors. To achieve the same upper speed limit of 2778 steps/sec, the single controlled motor requires acceleration of only 2E5 or 10% of that required for six directly controlled motors. This has a large effect on the choice and cost of the motors. In Fig. 2.5, Curve 1 shows the torque required to achieve the acceleration required in one motor control, while Curve 2 shows the one in six motor control.

When the motor speed is above ω_6, which is lower than ω_1, the torque will not be sufficient to achieve the speed change.

3) The speed resolution is reduced further. Compare the calculated values given in Tables 2.1 and 2.2. A speed of say 2020 steps/sec must be achieved by operating at the nearest attainable step rates. For direct control of one motor, the attainable rates are 2,000 or 2,041 steps/sec whereas for six motors, the rates are 1,389 or 2,778. The percentage speed change available at rates of 100 and 1000 steps/sec are 3.7% and 50% for six controlled motors compared to 0.1% and 1% for a single motor.

By using six microprocessors each to control one motor, these problems can be lessened but only with increased cost and bulk.

2.4 ACCURATE MULTIPLE MOTOR CONTROL

The solution to the problems mentioned above is to use one dedicated controller with microprocessor functions per motor so as to achieve the performance typified by the values in Table 2.1. The addition of an encoder disk permits closed loop control. In particular, the time between steps is then set by the movement of the rotor rather than by a timing loop. While the time taken for the rotor to move is not quantized, the time at which the encoder is sampled by the microprocessor is still quantized. However, closed loop control prevents the loss of synchronism so that the previous requirement that the velocity change occurs within 1 step can be relaxed and acceleration requirements reduced.

The control requirements for a single motor have been incorporated in special VLSI circuits from a number of manufacturers. The HCTL-1000 (c.f. Fig. 3.4) made by Hewlett Packard provides 24 bit position accuracy and is particularly well suited to the

general control of hydraulic cylinders and DC motors. Units such as the LM628 from National Semiconductor can give full 32 bit PID control of DC motors and hydraulic cylinders. Other units such as the Intel IDD008 or the cybernetics Micro systems CY512 provide direct open loop control of a step motor. The HP unit met the requirements for a general purposed Stewart platform controller and is readily available. The 24 bit position accuracy is sufficient for most purposes so the HP unit was selected to be used in this application.

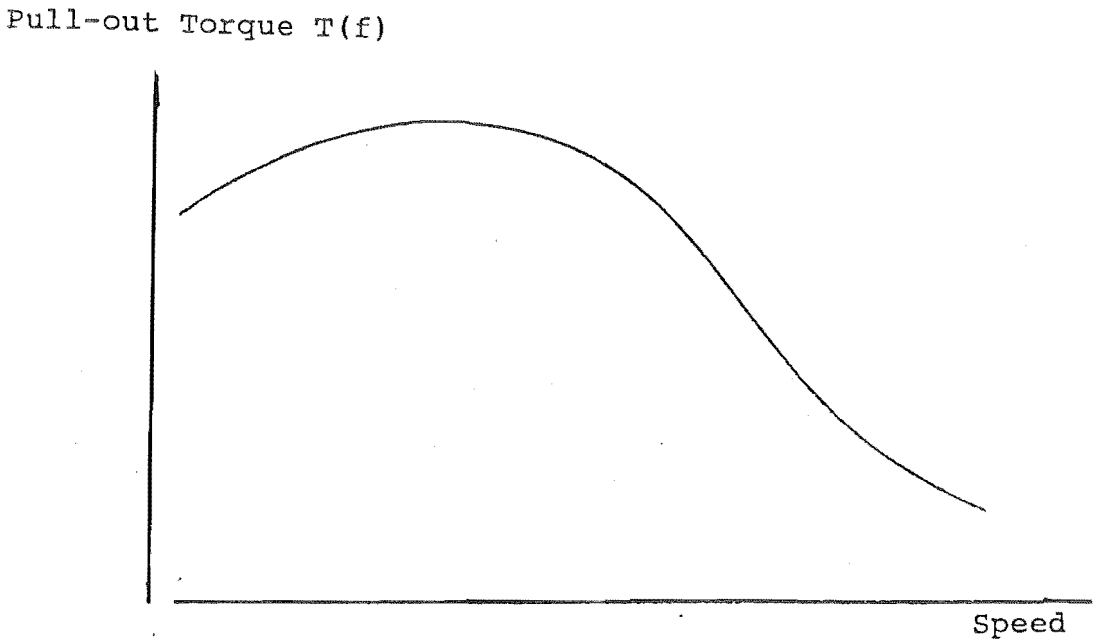


Fig. 2.1 Typical pull-out torque/speed characteristic

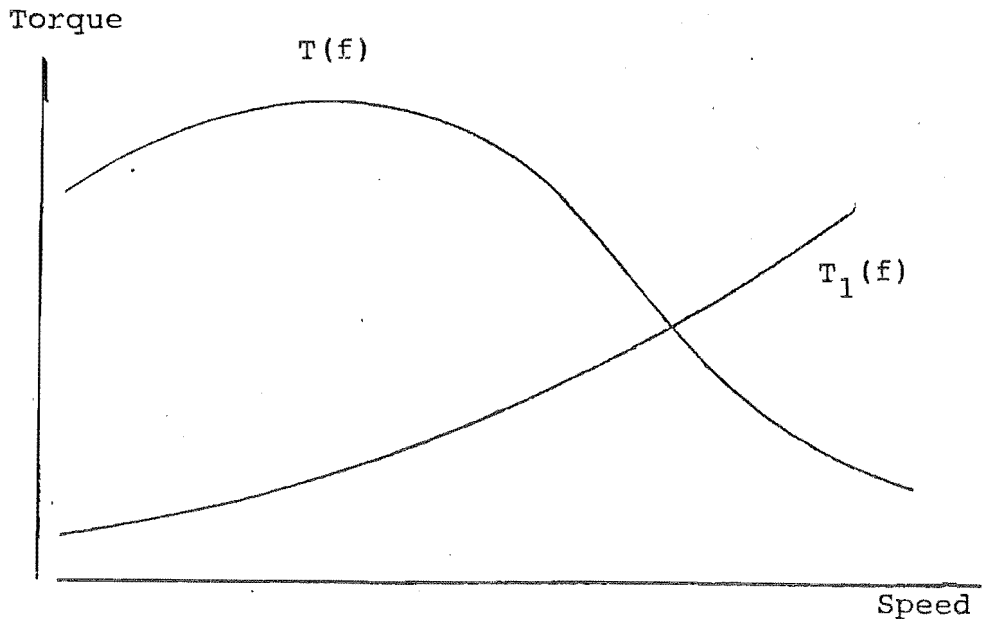


Fig. 2.2 Pull-out torque T(f) and load torque $T_1(f)$ characteristic

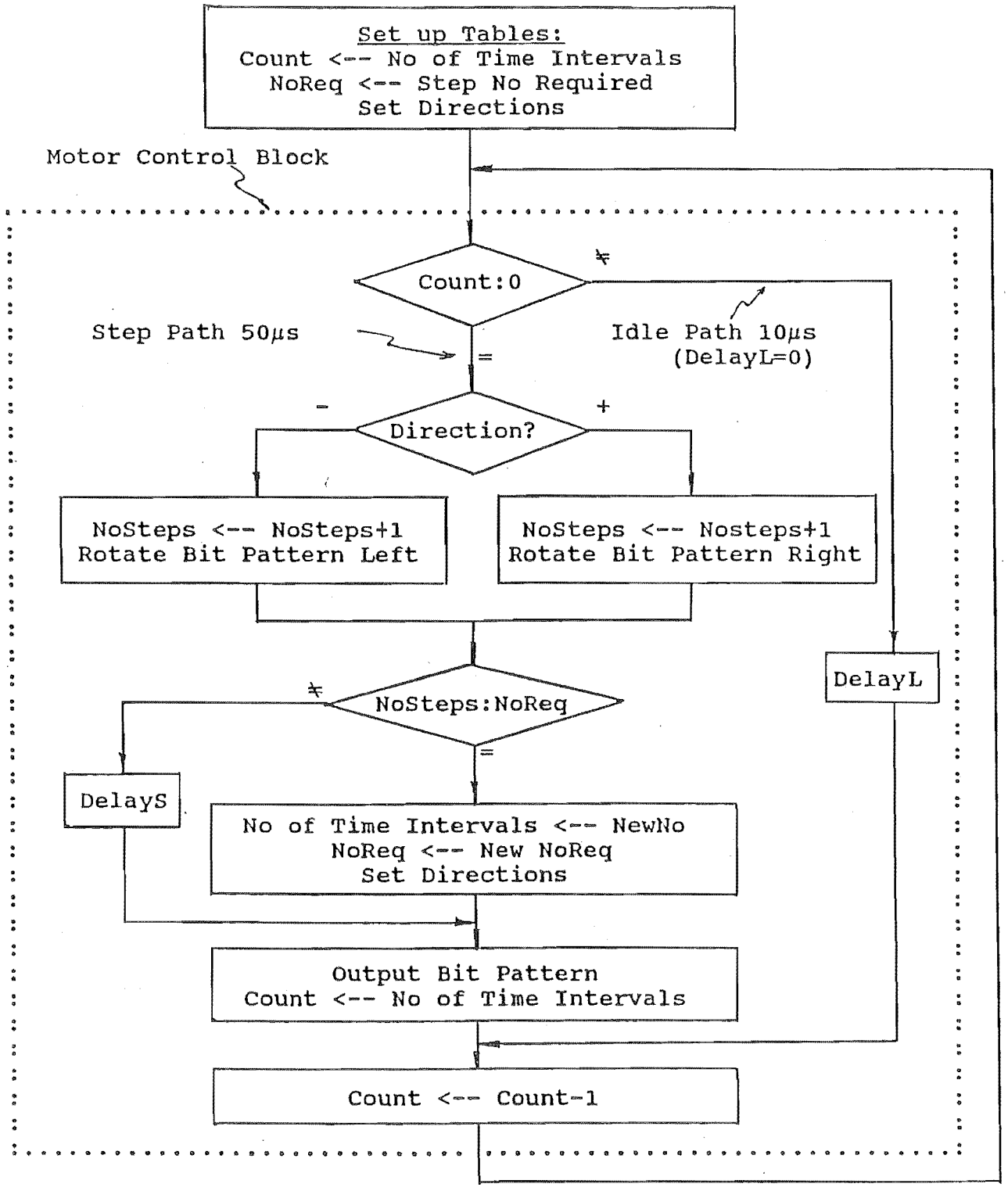


Fig.2.3 A typical flow chart for direct control of one motor.

Time Intervals	Loop Time µs	Step Rate steps/s	Speed Change steps/s	Acceleration steps/s/s
1	1*10+50=60	16667		
2	2*10+50=70	14286	2381.0	3.4E+07
3	3*10+50=80	12500	1785.7	2.2E+07
4	4*10+50=90	11111	1388.9	1.5E+07
5	5*10+50=100	10000	1111.1	1.1E+07
:	:	:	:	:
:	:	:	:	:
12	12*10+50=170	5882	367.6	2.2E+06
13	13*10+50=180	5556	326.8	1.8E+06
14	14*10+50=190	5263	292.4	1.5E+06
15	15*10+50=200	5000	263.2	1.3E+06
16	16*10+50=210	4762	238.1	1.1E+06
17	17*10+50=220	4545	216.5	9.8E+05
18	18*10+50=230	4348	197.6	8.6E+05
;	:	:	:	:
;	:	:	:	:
29	29*10+50=340	2941	89.1	2.6E+05
30	30*10+50=350	2857	84.0	2.4E+05
31	31*10+50=360	2778	79.4	2.2E+05
32	32*10+50=370	2703	75.1	2.0E+05
:	:	:	:	:
:	:	:	:	:
43	43*10+50=480	2083	44.3	9.2E+04
44	44*10+50=490	2041	42.5	8.7E+04
45	45*10+50=500	2000	40.8	8.2E+04
46	46*10+50=510	1961	39.2	7.7E+04
:	:	:	:	:
:	:	:	:	:
94	94*10+50=990	1010	10.3	1.0E+04
95	95*10+50=1000	1000	10.1	1.0E+04
96	96*10+50=1010	990	9.9	9.8E+03
:	:	:	:	:
:	:	:	:	:
994	994*10+50=9990	100	0.1	1.0E+01
995	995*10+50=10000	100	0.1	1.0E+01
:	:	:	:	:

Table 2.1 Typical step rates and accelerations produced by direct microprocessor control of one step motor. The algorithm shown in Fig.2.3 was used with DelayL=0µs

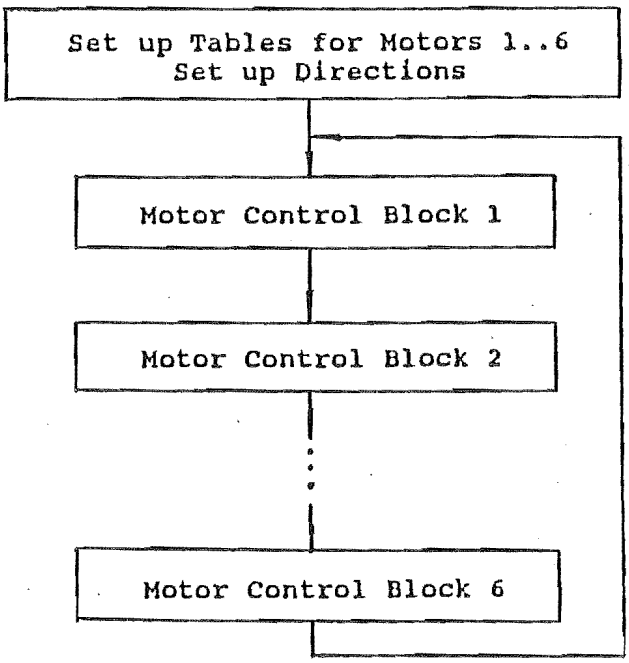


Fig. 2.4 A typical folwchart for direct control of six motors

Time Intervals	Loop Time Λs	Step Rate steps/s	Speed Change steps/s	Acceleration steps/s/s
1	1*60*6=360	2778		
2	2*60*6=720	1389	1388.9	1.9E+06
3	3*60*6=1080	926	463.0	4.3E+05
4	4*60*6=1440	694	231.5	1.6E+05
5	5*60*6=1800	556	138.9	7.7E+04
:	:	:	:	:
:	:	:	:	:
27	27*60*6=9720	103	4.0	4.1E+02
28	28*60*6=10080	99	3.7	3.6E+02
:	:	:	:	:

Table 2.2 Typical step rates and accelerations produced by direct microprocessor control of six step motors. The algorithms shown in Figs. 2.3 and 2.4 were used with DelayL = 50Λs

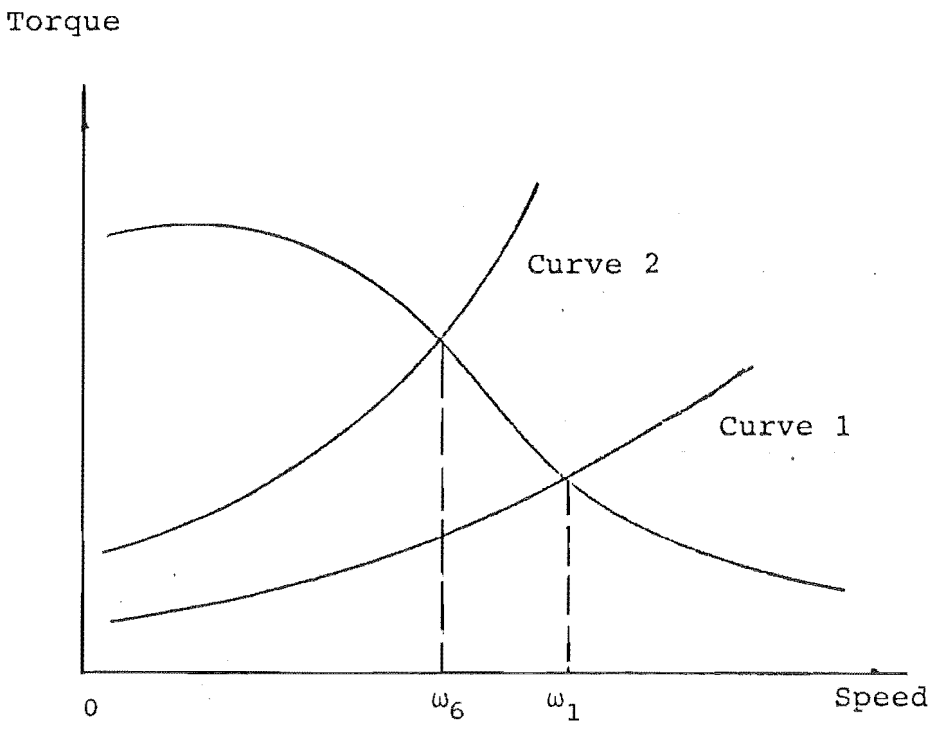


Fig. 2.5 Comparison of torque requirements for one and six direct motor control

CHAPTER 3

DESIGN OF A MULTI-MOTOR CONTROLLER

3.1 INTRODUCTION

Running multi-axis machinery, fulfilling the above task required and obtaining good performance at reasonable cost was the design principle of this whole system. The whole structure of the control system completed during this research is schematically shown in Fig. 3.1. The motor drives will be discussed in next chapter and the performance of the system is examined in CHAPTER 5. In this chapter, the design of a computer based six motor controller incorporating with six HCTL-1000 units is discussed.

In view of present conditions such as computers, HCTL-1000 units available commercially and future development, a versatile controller was designed. It can control one to six driving components. The driving components can be such as DC, DC brushless and step motors as well as hydraulic cylinders. The controller can be expanded to control more driving components with the same technique. The design will be described in three parts: computer, HCTL-1000 and the design of the controller.

3.2 THE COMPUTER AND ITS I/O CHANNEL

The IBM Personal Computer is used very popularly. It has Intel 8088 16-bit microprocessor as the Central Processor Unit. It has 16-bit registers and internal operations but uses an 8-bit input-output and data bus. The address bus is 20-bit so the maximum amount of memory that 8088 can address is 2^{20}, or 1MB of memory. It normally has 64 KB to 256 KB of RAM memory, 40KB ROM memory. It may have 8087 mathematics co-processor to increase the computing power of the machine. Also it has 8 expansion slots which allows additional devices to be added on the system. The 8088 runs on a 4.77 to 8 (8088-2 CPU) MHz clock cycle. It also has a proper and efficient Disk Operation System (DOS).

With so many features it was used as a card cage for the multi-motor controller Printed Circuit Board and a programmer for the HCTL-1000 motion control ICs. This desktop computer was chosen for the extensive expansion bus hardware and software support, as its memory and operation speed are enough in this application, plus its low cost.

The Input/Output (I/O) channel of an IBM-PC is an extension of its Intel 8088 microprocessor bus. It contains an 8-bit bidirectional data bus, a 20-bit address bus (the top 10 bits are not decoded), I/O read and write, and some memory control lines. These functions are provided via a 62-pin edge connector (c.f. Fig 3.2) which is also called "expansion slot". There are 8 slots on the system board of an IBM-PC. The I/O devices are addressed using I/O mapped address space. The channel is designed so that 768 I/O device addresses are available to the I/O channel cards. The prototype card, which is for users to design their own circuit to connect to the expansion I/O channel of the microprocessor bus, occupies 32 spaces from 300-31FH. The controller were allocated 12 (of 32) I/O addresses from 300H to 305H and from 308H to 30FH.

The functions of used I/O signals are:

i) A0-A11: Output. Can address 12 bit's I/O device. Logic high is 1.

ii) D0-D7: I/O. Provide data bus bits 0 to 7 for the microprocessor memory and I/O devices. Logic high is 1.

iii) -IOR: Output. -I/O Read Command: This command line instructs an I/O device to drive it's data onto the data bus. Active low.

iv) -IOW: Output. -I/O Write Command: This command line instructs an I/O device to read the data on the data bus. Active low.

v) AEN: Output. Address Enable. This line is used to de-gate the microprocessor and other devices from the I/O channel to allow DMA transfers to take place. Active high.

vi) RESET DRV: Output. Reset Drive: to reset or initialize system logic upon power-up. Active high.

3.3 THE HCTL-1000

The HCTL-1000 is a high performance, general purpose motion control IC. It performs all the time-intensive tasks of digital motion control, thereby freeing the host processor for other tasks. The complete servo system is made up of a host processor to specify commands, an HCTL-1000 to fulfill the algorithm, an amplifier and motor with an incremental encoder (c.f. Fig. 3.3).

3.3.1 The Operation

The HCTL-1000 provides position and velocity control for DC, DC brushless and step motors. Fig. 3.4 shows the internal block diagram of the HCTL-1000 unit. The HCTL-1000 receives its input commands from a host processor, here the IBM-PC, and position feedback from an incremental encoder with quadrature outputs. An 8-bit

bidirectional multiplexed address/data bus interfaces the HCTL-1000 to the host processor. The encoder feedback is decoded into quadrature counts and a 24-bit counter keeps track of position. The HCTL-1000 executes any one of four control algorithms selected by the user. The four control modes are:

-Position Control;

-Proportional Velocity Control;

-Trapezoidal Profile Control for point to point moves;

-Integral Velocity Control with continuous velocity profiling using linear acceleration.

More details are given in CHAPTER 5.

The resident Profile Generator calculates the necessary profiles for Trapezoidal Profile Control and Integral Velocity Control. The imbedded microprocessor in the HCTL-1000 compares the desired position (or velocity) to the actual position (or velocity) and computes the error which is input to a programmable digital filter $D_1(z)$. The filter output provides compensated motor control in the form of an 8 bit DAC voltage, or a Pulse Width Modulated (PWM) signal and direction signal. A commutator port can be programmed to provide electronic commutation for brushless DC and step motors. Thus, by using such a comprehensive microprocessor, the step time quantization problem can be greatly reduced.

3.3.2 User Accessible Registers

The HCTL-1000 operation is controlled by a bank of 64 8-bit internal registers, 32 of which are user accessible. These registers contain command and configuration information. Only by accessing these registers, can users run the controller chip to perform each required control functions. Fig. 3.5 shows the functional block diagram of the role of the user accessible registers.

3.3.3 Timing Diagram

The control bus of the HCTL-1000 contains four I/O lines, $\overline{ALE}$ (Address Latch Enable), $\overline{CS}$ (Chip Select), $\overline{OE}$ (Output Enable) and R/$\overline{W}$ (Read/Write). These lines execute the data transfers between the registers and the host processor over the 8-bit address/data multiplexed bidirectional bus. By satisfying a timing requirement, these lines can execute data transfers.

There are three different timing configurations. The $\overline{ALE}$/$\overline{CS}$ non overlapped timing configuration was used upon the use of 8088 microprocessor in IBM-PC (c.f. Fig. 3.6).

To access data to or from the HCTL-1000 registers, $\overline{ALE}$ (Address Latch Enable) pulse is asserted first. This routs the external bus data into an internal address latch (so called "external" and "internal" are relative to the HCTL-1000).

$\overline{CS}$ (Chip Select) low after rising $\overline{ALE}$ sends the data in the external bus into the data latch. Rising $\overline{CS}$ starts the internal synchronous process.

In the case of a write, the data in the data latch are written into the addressed location. In the case of read, the addressed location is written into an internal output latch, then $\overline{OE}$ (Output Enable) low enables the internal output latch onto the external bus. Thus the course of data transfers is completed.

3.4 DESIGN OF THE CONTROLLER

A six motor general purpose controller was designed during this research. The schematic logic diagram is shown in Fig. 3.7. More illustrations in detail for each part of it are given in the following subsections.

3.4.1 IBM-PC I/O Bus

The signals used to communicate with the HCTL-1000 units come from IBM-PC I/O bus (c.f. Fig. 3.2). The signals are provided in a 62-pin expansion slot sitting on the system board of IBM-PC. The functions of them were illustrated in 3.2.1.

3.4.2 Decoder and Control Logic

This part decodes all address and data accessing commands from the host processor and controls the correct timing.

3.4.2.1 Clock

The diagram of the external clock is shown in Fig. 3.8. A 3.6864 MHz crystal was used as an oscillator. A DM74109 Dual J-K Positive-Edge-Triggered Flip-Flop was used to generate square clock output at frequency of 1.8432 MHz. The clock output Pin 6 of U3 (abbreviated as P6/U3, same with others from now) goes to Pin 34 of each HCTL-1000 unit to generate the processor control timing and to be used when setting the sample period. Synchronism between the host and the slave processors is not implimental at this stage.

3.4.2.2 Control signals decoding

The four control signals $\overline{ALE}$, $\overline{CS}$, R/$\overline{W}$ and $\overline{OE}$ are decoded shown in Fig. 3.9 in details. U4 is a 74LS688 Magnitude Comparator which compares bit for bit two 8-bit words and indicates whether or not they are equal. One word is fixed as 30H by connecting each bit respectively to Vcc or GND. Another word is input from the microprocessor address bus A11-A4. When an I/O address having XX30XH pattern is input via the 20-bit I/O address bus (NB: X – not care), the middle part of it 30H will match the fixed word, so the Pin 19 of U4 is selected (goes low).

Note that the signal AEN (Address Enable) in IBM-PC I/O bus is used here to disable the Pin 1 of U4. The AEN is a DMA (Direct Memory Access) control signal. When this line is active (high), which happens regularly for memory refresh, the DMA controller will take over the control of all the address bus, data bus and control lines to allow DMA transfers to take place. Because DMA also communicates with I/O devices, the active high DMA interrupt regularly appears on the expansion bus. This will disturb the decoding logic of the controller. Thus AEN must be used so that no I/O addresses in the six motor controller can be selected when the AEN is taking place.

U5, U6 and U7 are 74LS138 Decoders/Demultiplexers which decode one-of-eight lines, based upon the conditions at the three binary select inputs (Pin 1-3) and the three enable inputs (Pin 4-6). IORD and IOWR from I/O bus are used directly or decoded through a series of gates to give R/$\overline{W}$ signal for the HCTL-1000.

If IOWR, address XX30XH (from Pin 19 of U4) and one of the addresses 8H to DH (from slot pins A0-A3) occur and are input into input pins of U7 respectively, U7 will provide an $\overline{ALE}$ output signal to Pin 38 of an HCTL-1000 unit. If IOWR or IORD, address XX30XH and one of the addresses 0H to 5H occur, U5 will provide $\overline{CS}$ to Pin 39 of an HCTL-1000 unit. If IORD, address XX30XH and one of the addresses

8H-DH are written, U6 will provide $\overline{OE}$ to Pin 40 of an HCTL-1000 unit. Thus the timing requirement by the HCTL-1000 is satisfied.

The addition of the capacitor C3 is to delay the falling edge of $\overline{OE}$ to ensure it happens a minimum period after the rising edge of $\overline{CS}$ as required, therefore, the data output from the internal location of the HCTL-1000 into its internal output latch are valid.

3.4.2.3 Data transfer

U8 is a 74LS245 Tri-state Octal Bus Transceiver. When Pin 19 goes low, which means one of the I/O addresses from 300H-30DH used is selected, the data on one side of the U8 can be transmitted to the other side depending on write or read.cycle If the data are read into IBM-PC, the Pin 1 goes low, the data on the HCTL-1000's data bus will be transmitted to the IBM-PC data bus. In the case of writing data to the HCTL-1000, Pin 1 keeps high, then the data on the IBM-PC I/O data bus (D0-D7 in slot) will be transmitted to the data bus of the HCTL-1000 units (Pin 2-9).

3.4.2.4 An example

An example is taken to explain the functions of the Decoder and Control Logic, combining with software program written in assembly language. A number 83H, for instance, is to be placed in the HCTL-1000's internal register R20H and a number is to be read from. Table 3.1 shows the whole procedure.

3.4.3 Reset

The Reset part is shown in Fig. 3.10. A RESET DRV signal (Pin B2 on slot) is led to Reset all HCTL-1000 units (Pin 36) upon power-up. A hardware manul reset switch

S1 is mounted. The Reset signal received by the HCTL-1000 will lead to a reset of internal circuitry and a branch to Reset mode.

3.4.4. HCTL-1000

Six HCTL-1000 units are used each for driving one motor. PWM and its SIGN signals are used for driving DC motors. Some provision were made for driving other types of driving components, i.e., PHA-PHD and $\overline{\text{INDEX}}$ signals are led (c.f. Fig. 3.11).

Except 4 control lines, 8 bit address/data, Clock, Reset are received by each HCTL-1000 to process the algorithm, total 12 signals are led from each HCTL-1000 unit. The 12 signals are:

i) PULSE (Pin 16) – Pulse Width Modulated signal. Output. The duty cycle is proportional to the Motor Command magnitude and is used for a DC motor or hydraulic cylinder speed control signal;

SIGN signal (Pin 17) – Output. Gives the sign/direction of the pulse signal

ii) PHA-PHD (Pin 26-29) – Output. Four commutator signals to provide phase switching information to step or DC brushless motors;

$\overline{\text{INDEX}}$ (Pin 33) – Index Pulse, input from the reference or index pulse of an incremental encoder. Used only in conjunction with the Commutator.

iii) INIT (Pin 13) – Initialization/Idle Flag. Output. This is a status flag indicating that the controller is in the Initialization/Idle mode.

iv) $\overline{\text{STOP}}$ (Pin 15) and $\overline{\text{LIMIT}}$ (Pin 14): Input. Internal flags set externally. A manul $\overline{\text{LIMIT}}$ switch is mounted beside the Stewart platform for use in the case of emergency (c.f. Fig. 3.1). Once the $\overline{\text{LIMIT}}$ flag is set in any control mode, it causes the HCTL-1000 to go into the Initialization/Idle Mode, clearing the Motor Command and causing an immediate motor shutdown.

v) CHA/CHB (Pin 31/30): Channel A, B – Input pins for position feedback from an incremental shaft encoder. Two channels A and B are 90° out of phase which will give the counter of the HCTL-1000 quadrature position input and also give direction information.

3.4.5. Buffer

Each of the Input/Output signals of the HCTL-1000 units is buffered by a channel of a 74LS244 Octal Tri-state Buffer, which provides improved noise rejection and high load drives (c.f. Fig. 3.12).

3.4.6 Controller Socket

A 96-pin connecter is soldered on the controller PCB, which is inserted on an expansion slot. The signals are shown in details in Fig. 3.12 and go to the motor drives through cables.

3.5 Testing and Debugging

A general purpose testing program was written and used to access every register inside the HCTL-1000 when debugging. The flowchart is shown in Fig. 3.13.and the program attached in APPENDIX C.

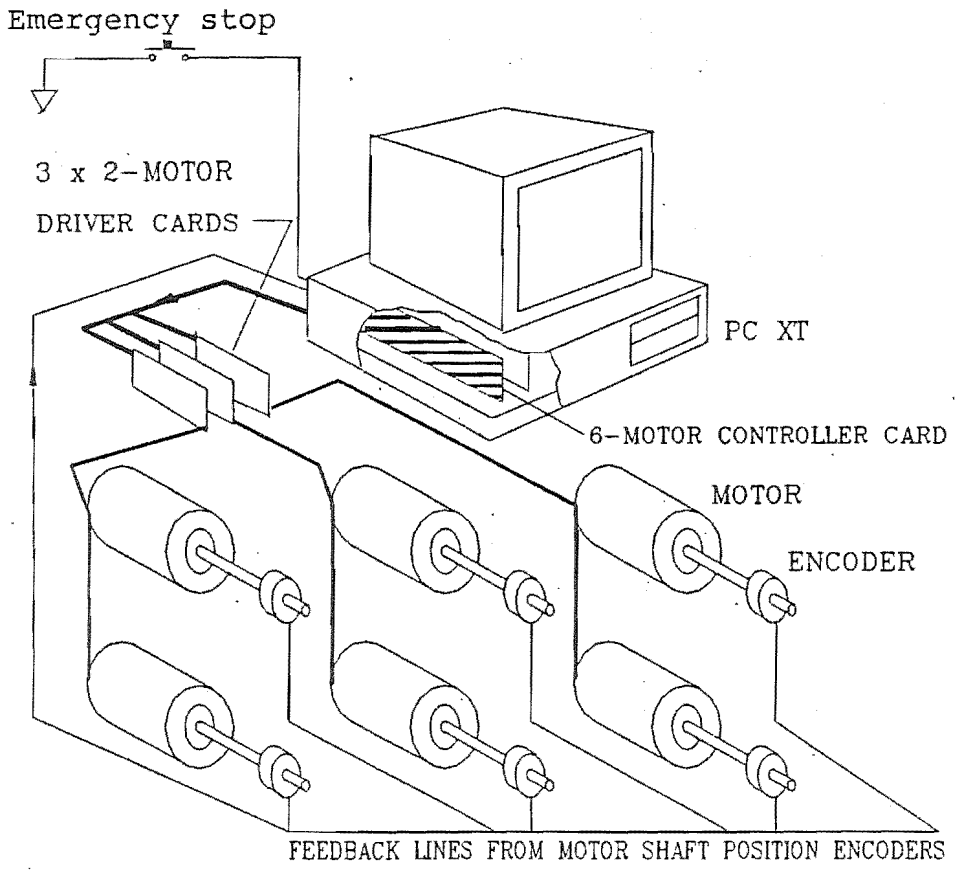


Fig. 3.1 Schematic structure of the control system for Stewart platform

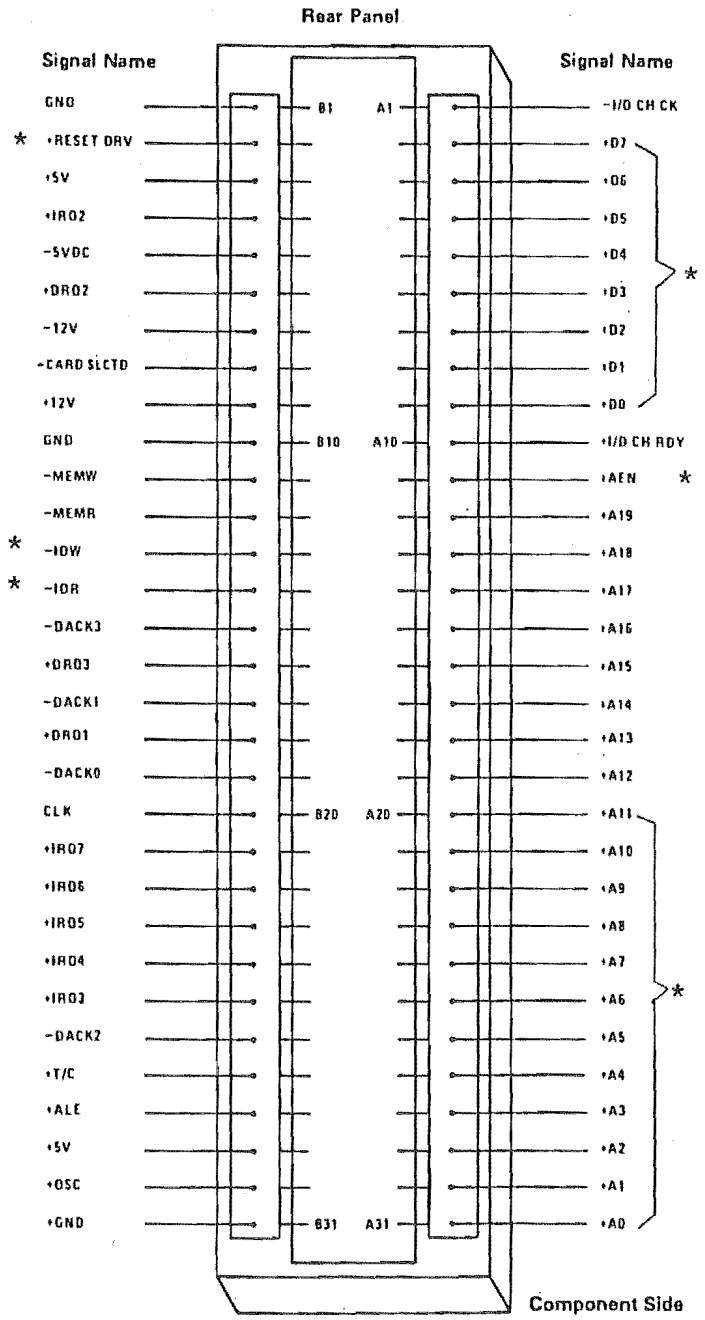


Fig. 3.2 IBM-PC computer I/O channel diagram
* signals used in the motor controller

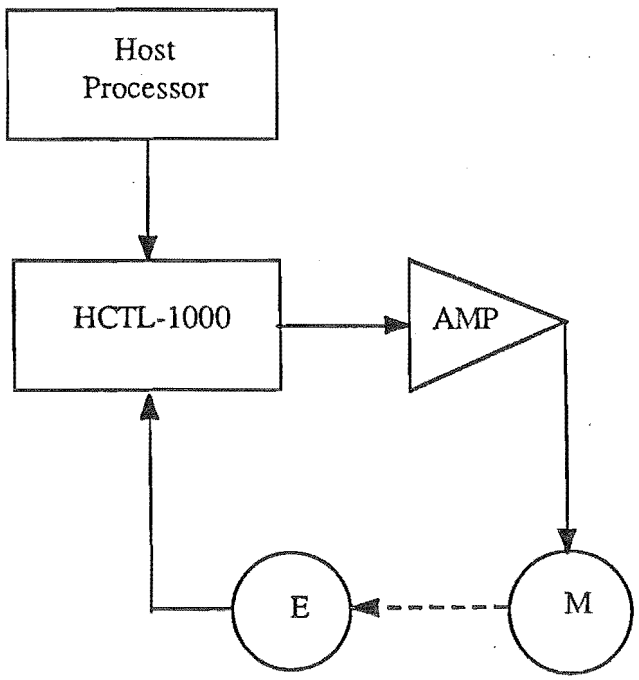


Fig.3.3 System block diagram of HCTL-1000

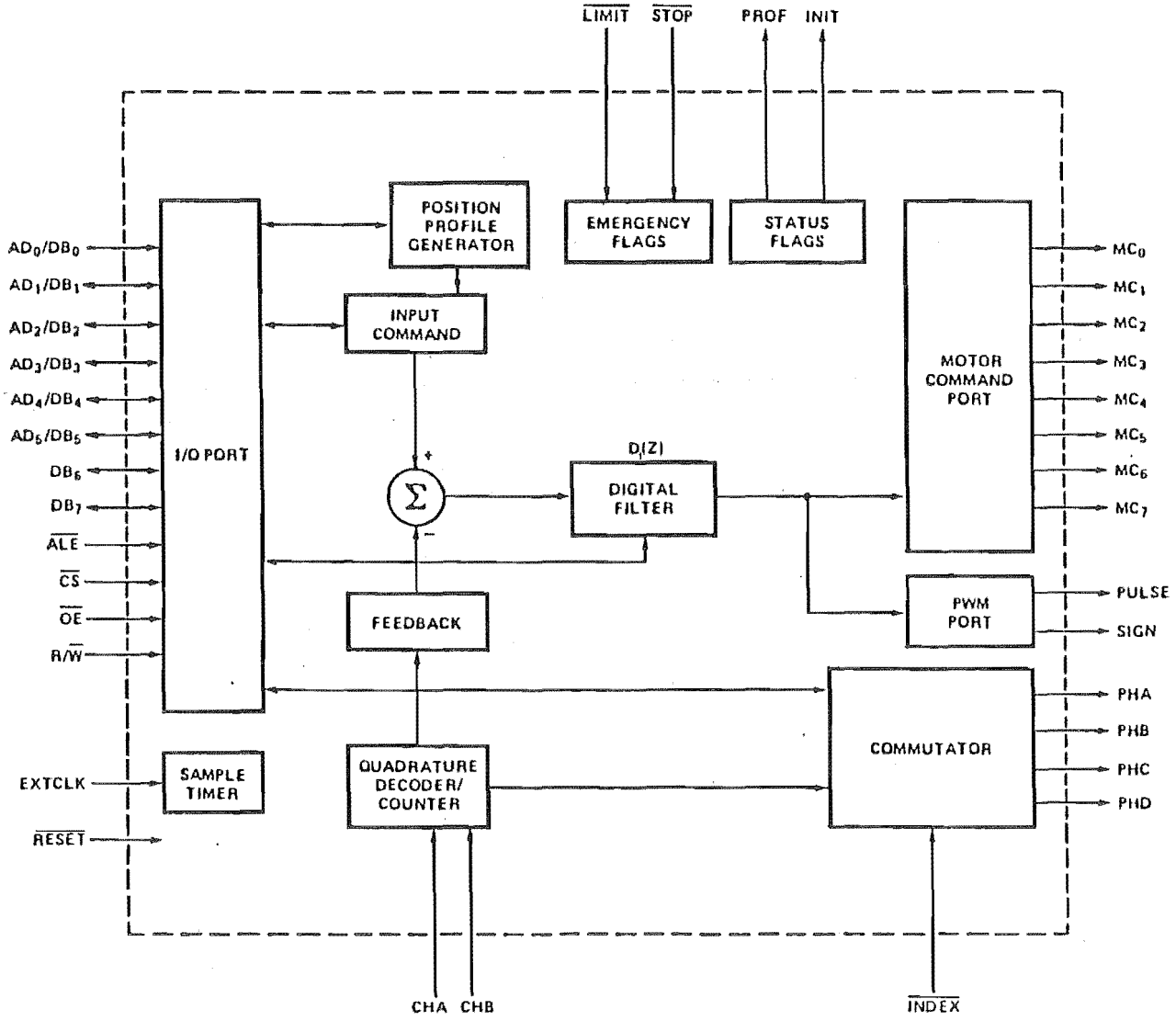


Fig. 3.4 HCTL-1000 internal block diagram

Fig. 3.5 The HCTL-1000 register functions

Register (Hex)	Function	Mode Used	Data Type	User Access
R00H	Flag Register	All	—	w
R05H	Program Counter	All	scalar	w
R07H	Status Register	All	—	r/w[1]
R08H	8 bit Motor Command Port	All	2's complement+80H	r/w
R09H	PWM Motor Command Port	All	2's complement	r/w
R0CH	Command Position (MSB)	Position Control	2's complement	r/w[2]
R0DH	Command Position	Position Control	2's complement	r/w[2]
R0EH	Command Position (LSB)	Position Control	2's complement	r/w[2]
R0FH	Sample Timer	All	scalar	w
R12H	Actual Position (MSB)	Position Control	2's complement	r[3]
R13H	Actual Position	Position Control	2's complement	r[3]/w[4]
R14H	Actual Position (LSB)	Position Control	2's complement	r[3]
R18H	Commutator Ring	All	scalar[5]	r/w[6]
R19H	Commutator Velocity Timer	All	scalar	w
R1AH	X	All	scalar[5]	r/w
R1BH	Y Phase Overlap	All	scalar[5]	r/w
R1CH	Offset	All	2's complement	r/w[6]
R1FH	Maximum Phase Advance	All	scalar[5]	r/w[6]
R20H	Filter Zero, A	All except Proportional Velocity	scalar	r/w
R21H	Filter Pole, B	All except Proportional Velocity	scalar	r/w
R22H	Gain, K	All	scalar	r/w
R23H	Command Velocity (LSB)	Proportional Velocity	2's complement	r/w
R24H	Command Velocity (MSB)	Proportional Velocity	2's complement	r/w
R26H	Acceleration (LSB)	Integral Velocity and Trapezoidal Profile	scalar[5]	r/w
R27H	Acceleration (MSB)	Integral Velocity and Trapezoidal Profile	scalar[5]	r/w
R28H	Maximum Velocity	Trapezoidal Profile	scalar[5]	r/w
R29H	Final Position (LSB)	Trapezoidal Profile	2's complement	r/w
R2AH	Final Position	Trapezoidal Profile	2's complement	r/w
R2BH	Final Position (MSB)	Trapezoidal Profile	2's complement	r/w
R34H	Actual Velocity (LSB)	Proportional Velocity	2's complement	r
R35H	Actual Velocity (MSB)	Proportional Velocity	2's complement	r
R3CH	Command Velocity	Integral Velocity	2's complement	r/w

Notes:

1. Upper 4 bits are read only.
2. Writing to R0EH (LSB) latches all 24 bits.
3. Reading R14H (LSB) latches data into R12H and R13H.
4. Writing to R13H clears Actual Position Counter to zero.
5. The scalar data is limited to positive numbers (00H to 7FH).
6. The commutator registers (R18H, R1CH, R1FH) have further limit which are discussed in the Commutator section of this data sheet.

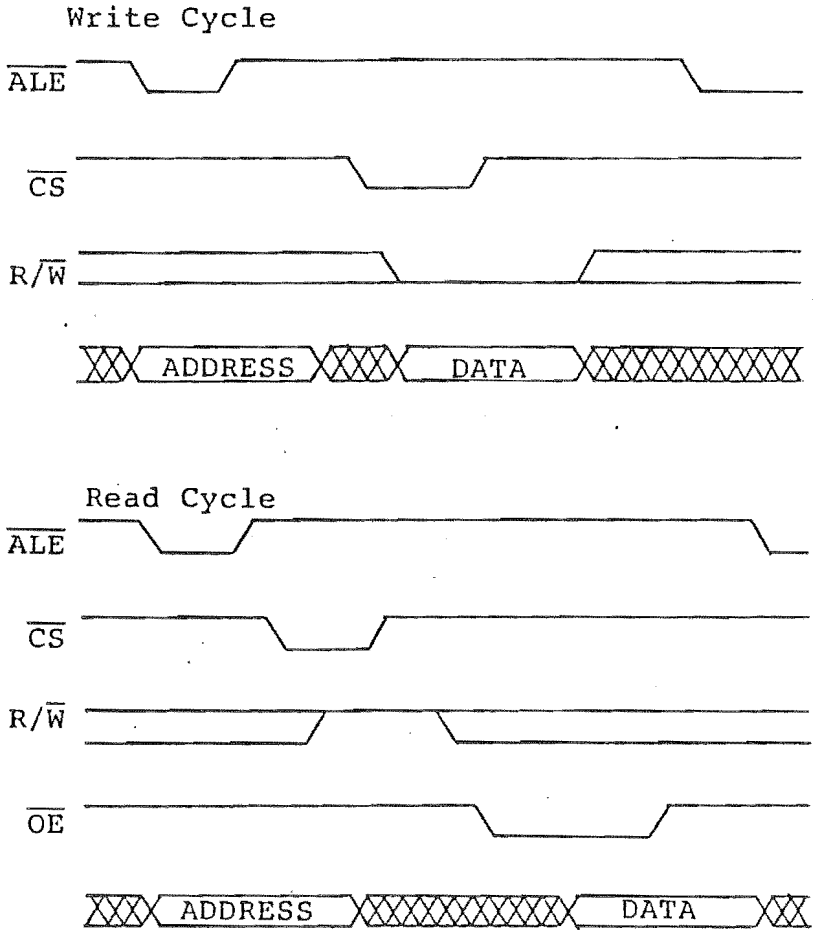
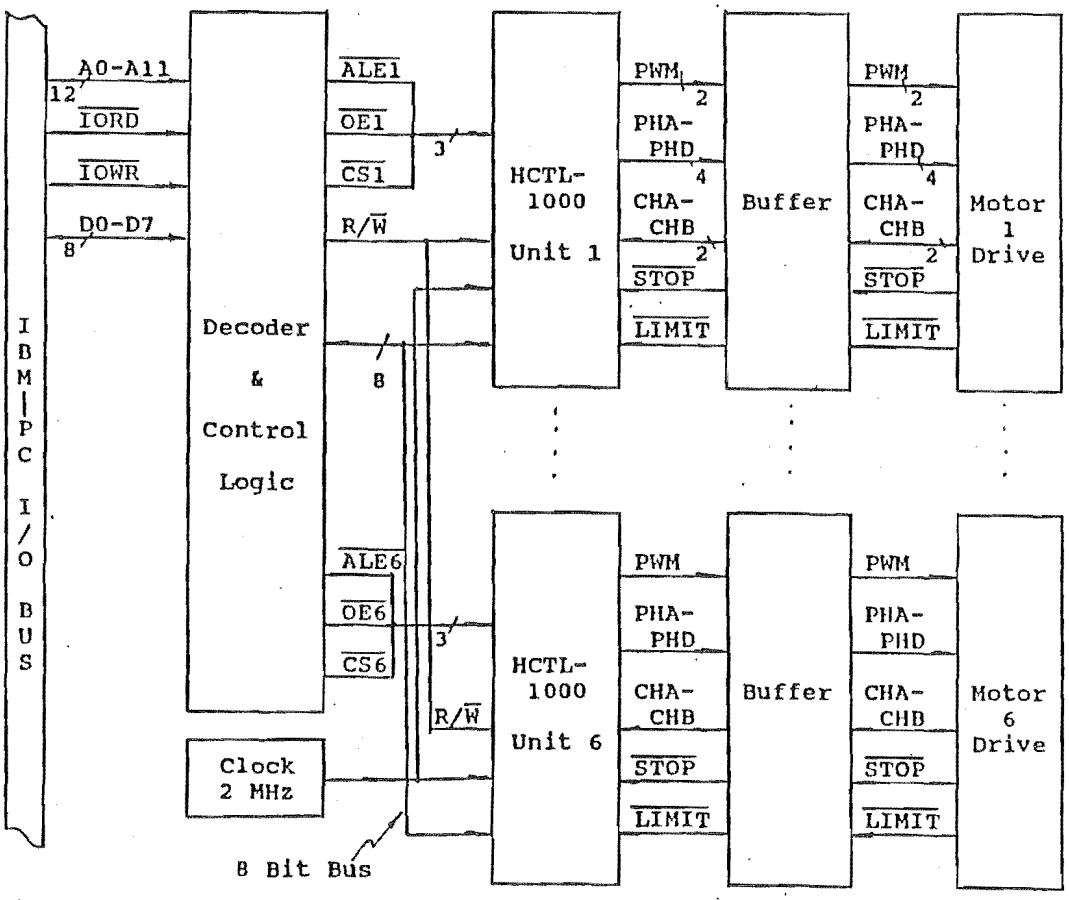


Fig. 3.6 HCTL-1000 ALE/CS non overlapped read and write timing



Each chip is accessed by 3 decode lines $\overline{ALE}$, $\overline{OE}$ and $\overline{CS}$.
The IBM-PC Prototype card addresses used are:

300H $\overline{CS1}$, read or write Unit 1;
:
305H $\overline{CS6}$, read or write Unit 6.

308H $\overline{ALE1}$, write the register number selected to Unit 1;
:
30DH $\overline{ALE6}$, write the register number selected to Unit 6.

308H $\overline{OE1}$, read the output buffer of Unit 1;
:
30DH $\overline{OE6}$, read the output buffer of Unit 6.

Fig. 3.7 Computer and motor interfaces for the motor controller

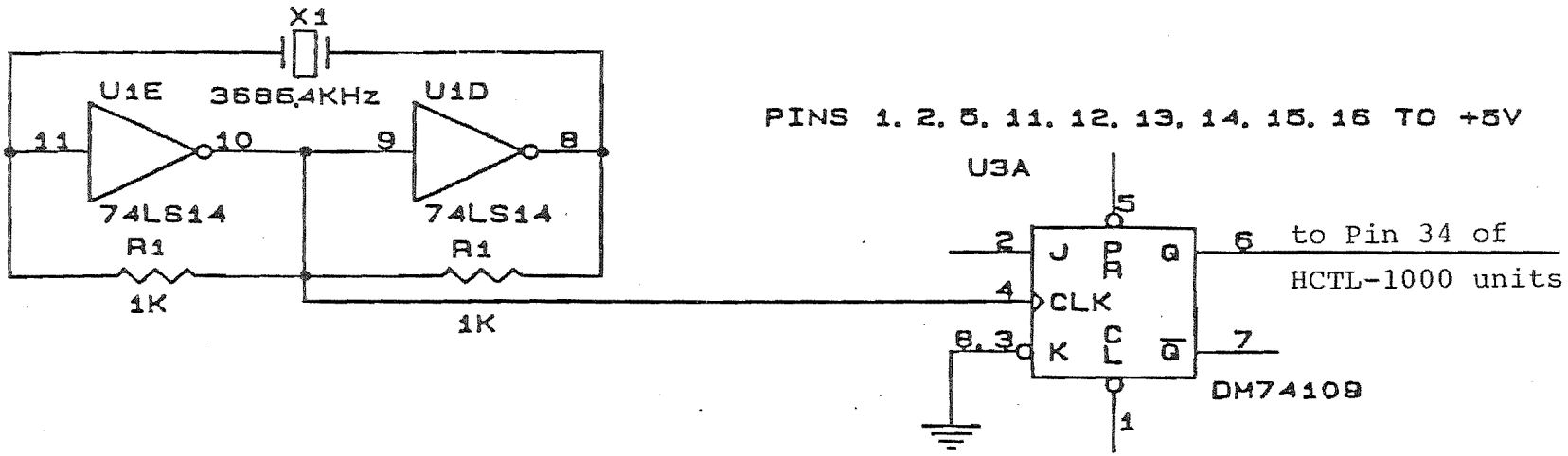


Fig. 3.8 Clock diagram for the motor controller

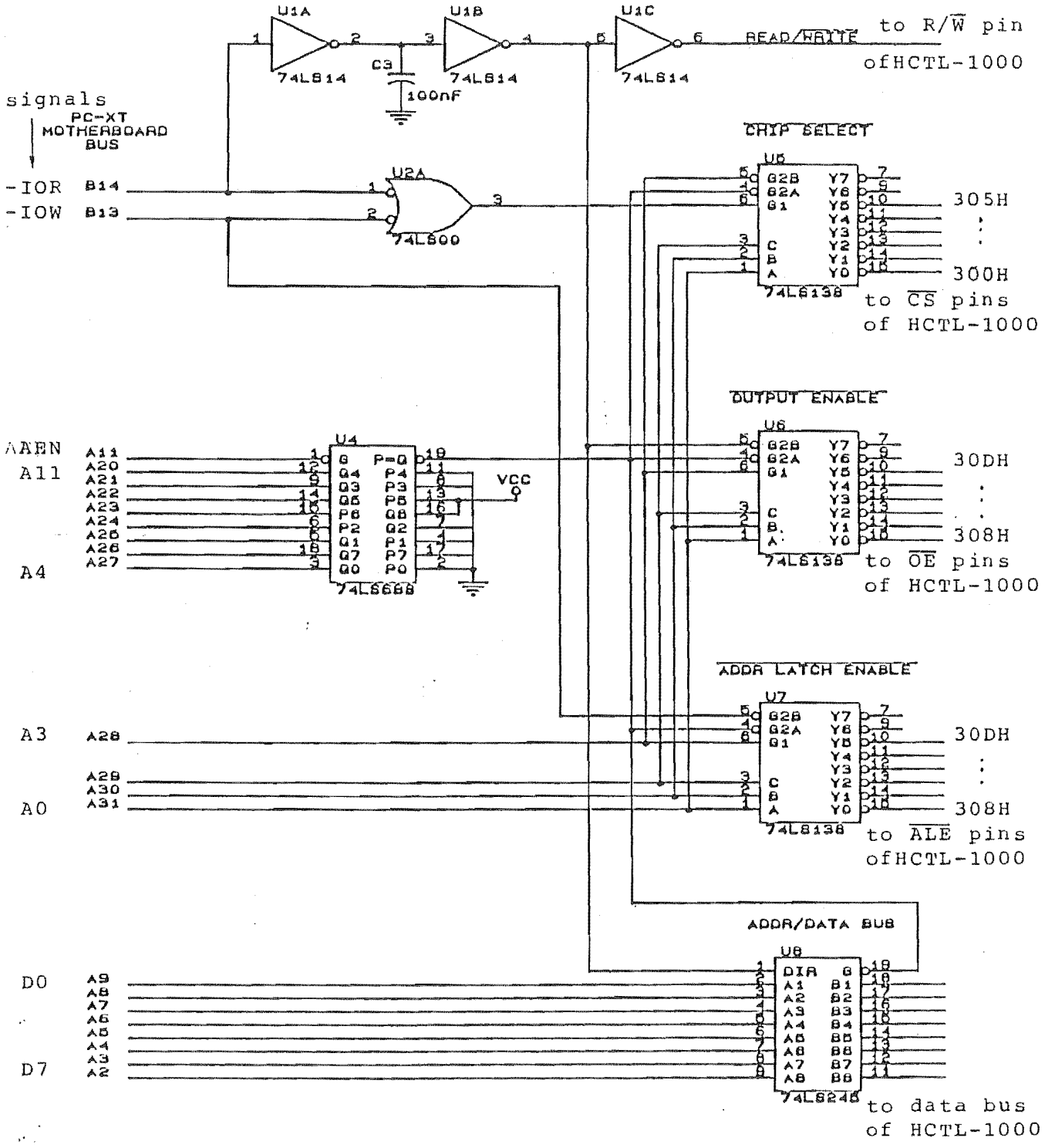


Fig. 3.9 Decoder and Control Logic of
the motor controller

Table 3.1 An Example - Write Data to and Read from HCTL-1000

Function	Program	IBC-PC	Interface	HCTL-1000 U14
Write 83H into register 20H in HCTL-1000 U 14	LD A, 20H OUT (308H), A	IOWR is generated on I/O bus, Address 308H appears on I/O bus A0 - A11;	$\overline{ALE}$ (on Pin 15 of U7) occurs	$\overline{ALE}$(on Pin 38) is given
		20H appears on I/O bus D0 - D7		which enables low 6 bits (20H on Pin 2-7) of external data bus into internal address latch
	LD A, 83H OUT (300H), A	IOWR is generated on I/O bus, Address 300H appears on I/O bus A0-A11;	$\overline{CS}$ (on Pin 15 of U5) occurs	$\overline{CS}$ (on Pin 39) is given, R/$\overline{W}$ (on Pin 37) is given
		83H appears on I/O bus D0-D7		The external data bus data (83H on Pin 29) is written into the internal address location

(Continued on next page)

(Table 3.1 continued)

Read a datum from register 20H in HCTL-1000 U14	LD A, 20H OUT (308H), A	IOWR is generated on I/O bus, Address 308H appears on I/O bus A0-A11;	$\overline{ALE}$ (on Pin 15 of U7) occurs	$\overline{ALE}$ (on Pin 38) is given,
		20H appears on I/O bus D0-D7		Which enables low 6 bits (20H on Pin 2-7) of external data bus into internal address latch
	IN A, (300H)	IORD is generated on I/O bus, Address 300H appears on I/O bus A0-A11;	$\overline{CS}$ (on Pin 15 of U7) occurs	$\overline{CS}$ (on Pin 39) is given, R (on Pin 37) is given, Data is read from an internal location into an internal output latch
	IN A, (308H)	IORD is generated on I/O bus, Address 308H appears on I/O bus A0-A11	$\overline{OE}$ (on Pin 15 of U6) occurs	R on Pin 37) is given, $\overline{OE}$ (on Pin 40) is given which enables the data in the internal output latch onto the external data bus to complete a Read operation
		The data to be read is on I/O bus D0-D7		

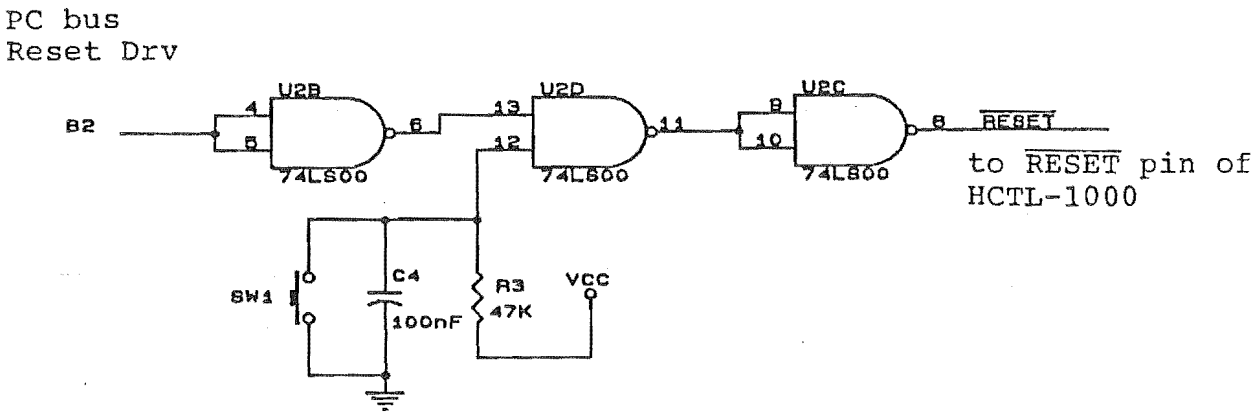


Fig 3.10 Reset diagram of the motor controller

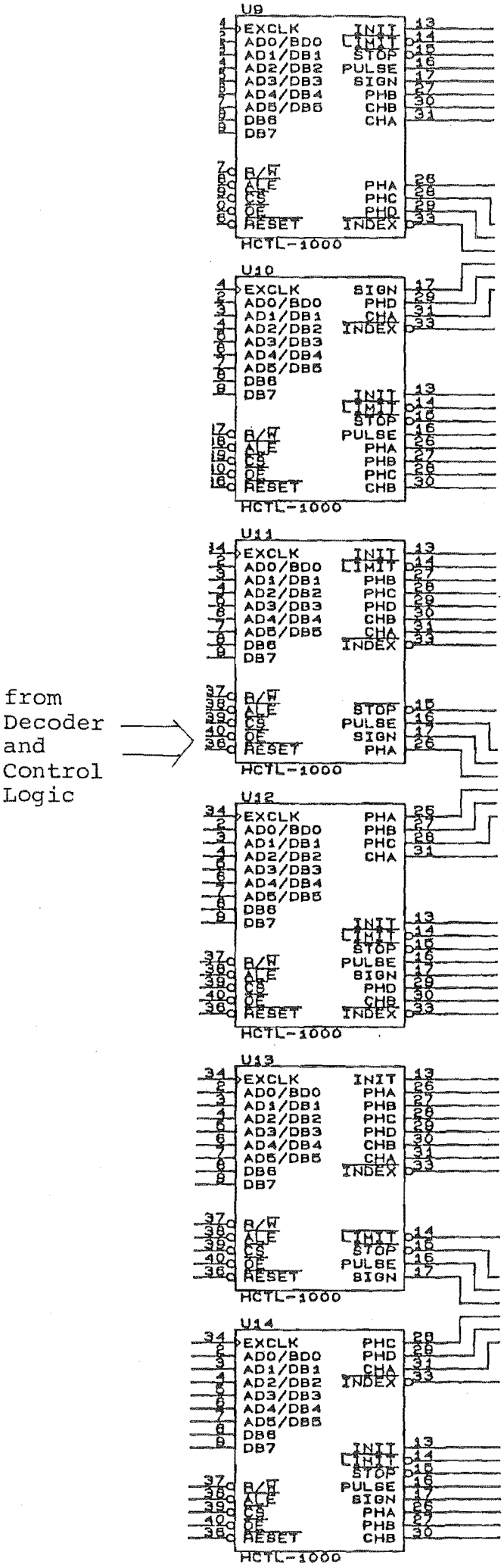
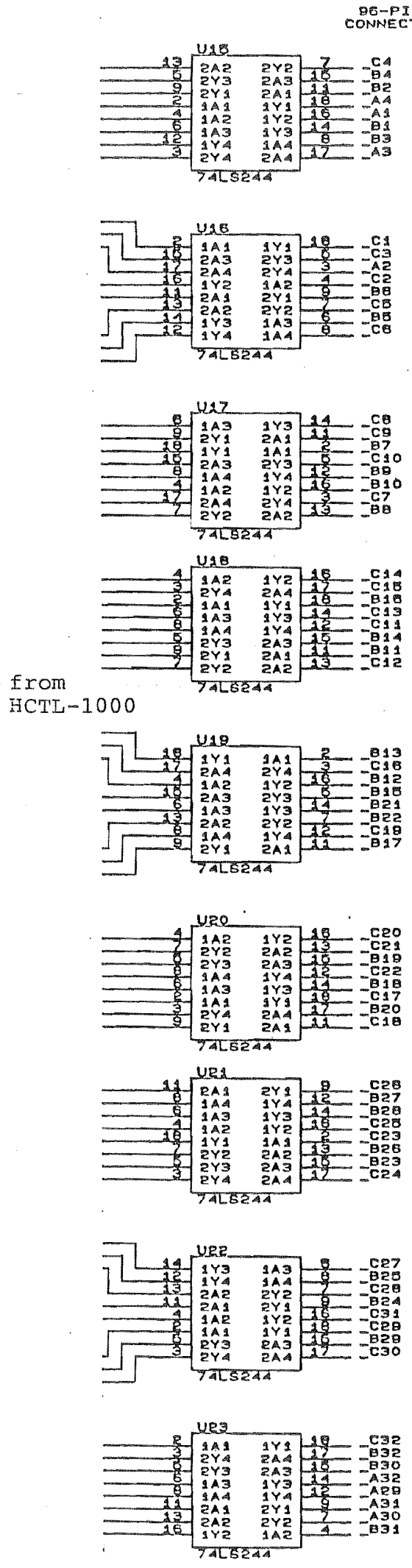


Fig. 3.11
HCTL-1000 input and output signals



	C	B	A	
FROM U9	PHA	PHB	SIGN	1
	INDEX	STOP	PHD	2
	PHC	CHB	CHA	3
	INIT	LIMIT	PULSE	4
FROM U10	PHD	CHA	+5V	5
	INDEX	SIGN	+5V	6
	PHC	STOP	+5V	7
	INIT	CHB	+5V	8
	LIMIT	PHA	+5V	9
	PULSE	PHB	+5V	10
FROM U11	PHD	CHA	+5V	11
	INDEX	SIGN	+5V	12
	PHC	STOP	+5V	13
	INIT	CHB	+5V	14
	LIMIT	PHA	+5V	15
	PULSE	PHB	+5V	16
FROM U12	PHD	CHA	GRND	17
	INDEX	SIGN	GRND	18
	PHC	STOP	GRND	19
	INIT	CHB	GRND	20
	LIMIT	PHA	GRND	21
	PULSE	PHB	GRND	22
FROM U13	PHD	CHA	GRND	23
	INDEX	SIGN	GRND	24
	PHC	STOP	GRND	25
	INIT	CHB	GRND	26
	LIMIT	PHA	GRND	27
	PULSE	PHB	GRND	28
FROM U14	PHD	CHA	SIGN	29
	INDEX	STOP	PHB	30
	PHC	CHB	PHA	31
	INIT	LIMIT	PULSE	32
	C	B	A	

PIN ASSIGNMENT
96-PIN CONNECTOR ON PCB

Fig. 3.12 Buffer and controller socket diagram

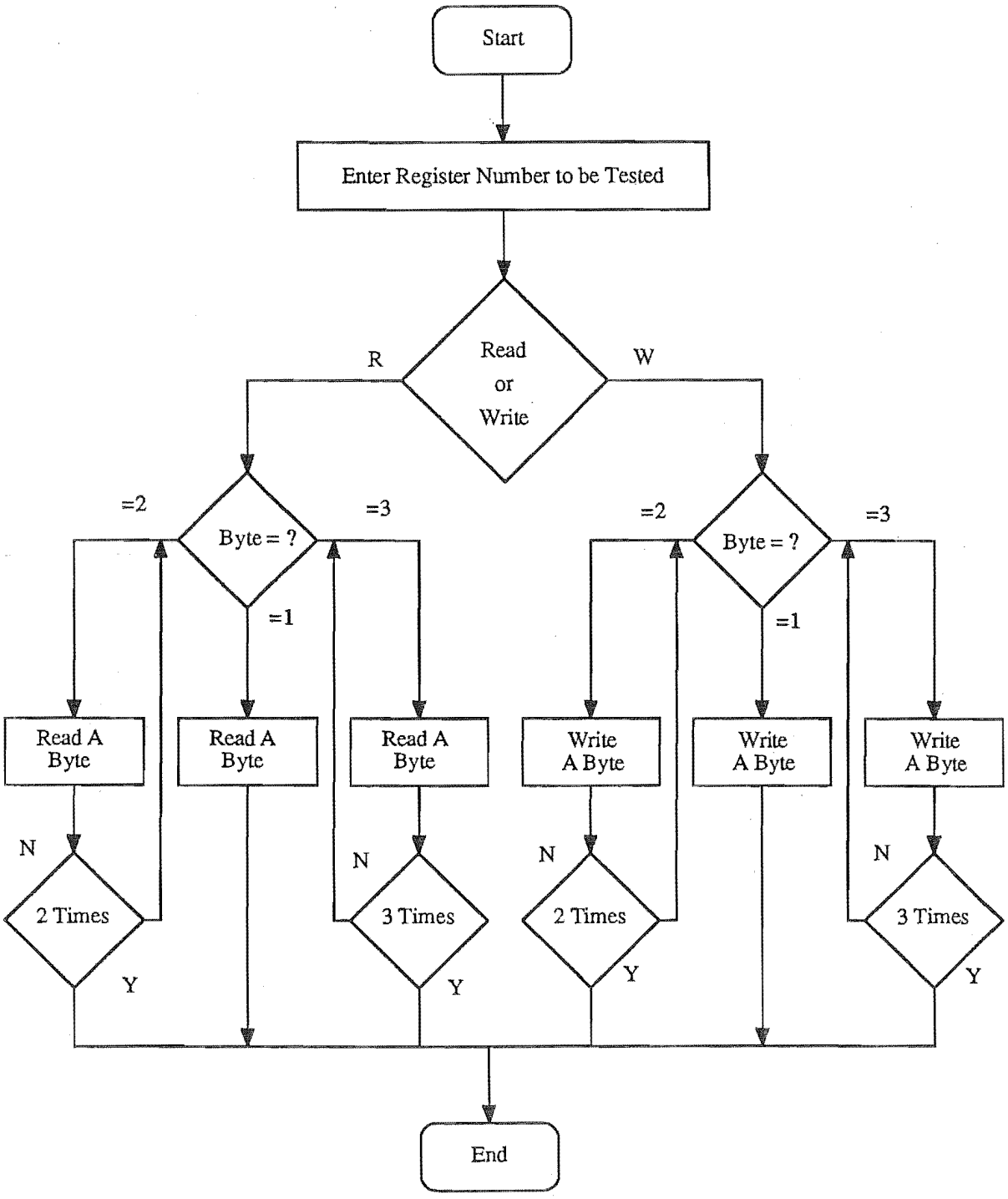


Fig. 3.13 Software flowchart for Read/Write registers in the HCTL-10000

CHAPTER 4

DC MOTOR PWM SERVO DRIVE

4.1 INTRODUCTION

The Stewart platform proposed in CHAPTER 1 has six legs driven by six actuators. Several types of actuators, such as hydraulic cylinders or various types of electric motors, could be chosen. Hydraulic cylinder linear actuators, which convert hydraulic power into linear mechanical work, are the best choice in this high precision and heavy duty control application due to their fast dynamic response, good mechanical stiffness, large force capability and high power. Unfortunately, they are also expensive and thus were not used at this stage of the project.

The DC motors became the next best choice owing to their outstanding advantages: They provide a wide variety of operating characteristics obtained by selection of the excitation method of the field windings. Also they provide ease of speed control, high efficiency, the conditions of high torque and low voltage, and are relatively inexpensive. In this research, six Electrak 100 Linear Actuators driven by permanent magnetic DC motors were chosen (c.f. Warner Electric Catalogue 1985).

The principle of operation of the actuators is shown in Fig. 4.1. A drive screw and a recirculating ball covert the rotating movement, transmitted from the motor via the gear train, into linear movement, so that the extension tube (the leg of the Stewart platform), will expand or contract as required for the platform motion.

The parameters of the Electrak 100 linear actuator are:

Stroke length	24"
Pitch of the screw	0.2"
Gear train:	N1= 16
	N3= 80
Gear train ratio	1:5

The permanent magnet DC motor in the actuator is shown in Fig. 4.2 (c.f. Warner 1985). There is a thermal protector inside of the motor housing, which protects the motor from overheating.

The overview of various DC motors, particularly the permanent magnetic DC motors, which has good characteristics for speed control, will be given in next section.

4.2 DC MOTOR

4.2.1 Operation Principles

The DC motor is basically a torque transducer. The torque is produced by the net force from all conductors acting over an average radial length to the shaft center (c.f. Fig. 4.3). The DC motor's characteristic is expressed in a linear relationship: the mechanically developed shaft torque of a DC motor is directly proportional to its armature current as the following:

$$T_m = K_p \Phi I_a \qquad (4.1)$$

Where, T_m – the shaft torque in Newton-meters;

Φ – the magnetic flux per pole in webbers;

I_a – the armature current in amperes;

K_p – a proportionality constant fixed by the design of the winding.

When the conductor moves in the magnetic field, an induced voltage or back electromotive force (emf) is generated across it, tending to oppose the current flow through the conductor. This voltage is proportional to the shaft velocity:

$$E = K_p \Phi \omega \qquad (4.2)$$

Where E – back emf in volts;

ω – velocity in radian/second.

In a motor, the back-emf voltage E, plus the voltage drop $I_a R_a$through the armature due to armature current I_a and armature resistance R_a, must be overcome by the total impressed voltage V_m at the terminals. The voltage relations are expressed by Eq. (4.3):

$$V_m = E + I_a R_a \qquad (4.3)$$

Where, V_m – terminal voltage in volts;

R_a – armature resistance in ohms.

Equations (4.1), (4.2) and (4.3) form the fundamental basis for DC motor operation (c.f. Fitzgerald 1983).

4.2.2 Classification and Characteristics

DC motors may be categorized as shunt, series, compound and separately excited depending on the method they employ to create the magnetic field (c.f. Fig. 4.4).

4.2.2.1 Shunt motor

The field circuit and the armature circuit of a DC shunt motor are connected parallel (c.f. Fig. 4.4 (a)). The field current and pole flux are essentially constant and independent of the armature requirements. The torque is therefore essentially proportional to the armature current (c.f. Eq. (4.1)).

In operation, decreases in speed and back emf E, due to added load, produces an increase in the small portion of voltage drop $I_a R_a$ (c.f. Eq. (4.3)). Since terminal voltage V_m is constant, I_a must increase some hence the output torque increase some to give extra torque required. This produces a wide range of speed and torque characteristics and the speed-load curve is practically flat (c.f. Fig. 4.5), resulting in the term "constant speed" for the shunt motor. Typical applications are for load conditions of fairly constant speed.

4.2.2.2 Series motor

The field circuit and the armature circuit of a DC series motor are connected in series (c.f. Fig. 4.4 (b)). The flux of a series motor is nearly proportional to the armature current I_a which produces it. As a result, the torque of a series motor is proportional to the square of the armature current, visa versa, large increase in torque may be produced by a relatively small increase in armature current (c.f. Eq. (4.1)).

In operation, the increase in load causes a decrease in rotor speed, hence a decrease in the back emf E. Due to decreased E, the armuture current must increase to provide more torque to drive the increased load. This produces a variable speed characteristic. The speed-load curve is the markedly dropping one shown in Fig. 4.5. The applications include heavy torque overloads, high starting torques and variable speeds.

4.2.2.3 Compound motor

A compound motor has two field windings. One is connected in parallel with the armature circuit; the other is connected in series with the armature circuit (c.f. Fig. 4.4(c)). With both field windings, this motor combines the effects of the shunt and series types to an extent dependent upon the degree of compounding. This motor is used when high starting torque and somewhat variable speeds are required.

4.2.2.4 Separately excited motor

The field winding of this motor is energized from a source different from that of the armature winding (c.f. Fig. 4.4 (d)), so that the two impressed voltages can be varied independently, thus produces a wide range of speed and torque characteristics. The required field current is usually a very small fraction of the rated armature current (Fitzgerald 1983). A small amount of power in the field circuit may control a relatively large amount of power in the armature circuit. Separately excited generators are often used in feedback control systems when control of the armature voltage over a wide range is required.

4.2.2.5 Constant field and permanent magnetic DC motor

Above four categories may all have variable magnetic flux. If the field excitation of a separately excited field motor is constant, the motor becomes a constant magnetic flux one. More particularly, when the motor is small, it can be made with permanent magnet field with armature excitation only, which produces constant flux. The permanent magnet field motor (c.f. Fig. 4.2) is described and highlighted in this separate subsection due to it's use in this project.

During the operation of a separately excited constant field motor or a permanent magnet field motor, the field flux is constant and Eq. (4.1) simply becomes

$$T_m = K_t I_a \tag{4.4}$$

Where, K_t is the torque constant of the motor;

and Eq. (4.2) becomes:

$$E = K_e \omega \tag{4.5}$$

Where, K_e – back emf voltage constant.

Note that, the mechanically developed power P_m must be equal to the electrical power absorbed in the rotor P_e, i.e.,

$$P_m = T_m \omega = K_t I_a \omega \tag{4.6}$$

$$P_e = E I_a = K_e \omega I_a \tag{4.7}$$

$$P_m = P_e \tag{4.8}$$

Therefore, in MKS system, $K_t = K_e$.

As can be seen, the relationship between armature current, torque and velocity is simply linear. This simple mathematical relationship gives the ease of control in incremental motion servo systems. However, the permanent magnet field motor has more advantages over the wound field structure one. Firstly, there is no power dissipated in the permanent magnet field. Therefore, the permanent magnet field motor is more efficient and requires less space. Secondly, due to the availability of permanent magnets with high coercive force characteristics, the linear relationship of Eq. (4.1) holds for higher armature currents in permanent magnet DC motor than in

their wound field DC motor counterparts. Thirdly, the more simple structure of permanent magnet field motor has more simple relationship between electrical input parameters and mechanical output characteristics. This is why the permanent magnet DC motor is very popular for use in motion control servo systems. The servo calculations and system design of this application are discussed in CHAPTER 5.

4.2.2.6 Motor parameters

Other parameters and characteristics of this permanent magnet motor in Electrak 100 are:

- Current Draw: 9.1 Amps at 24VDC – 2.2kN (500 lb) capacity at full load. Fig. 4.6(a) shows the specification of the relationship between current drawn and load capacity (c.f. Warner Catalogue).
- The specification of the tube extension speed vs load is shown in Fig. 4.6(b).
- Duty Cycle: 25% "on" time at maximum rated load per cycle.
-Measured resistance is 0.9 Ω.
-Measured inductance is 12mH.

The relationships between voltage, current and angular velocity are measured under the condition of non-load and shown in Fig. 4.7. It can be seen that the relationship is essentially linear. When the impressed voltage increases, the angular velocity will have a proportional increase while the current remains almost constant provided that the load is unchanged. Thus, the motor back emf constant K_e can be obtained (c.f. Eqs.(4.3) and (4.5)):

$$K_e = 0.073 \text{ (volts.s/radian)}$$

4.3 DC MOTOR PWM SERVO DRIVE

4.3.1 Introduction

For driving a DC servo system, either a voltage or a PWM control signal can be used to determine the correct amount of current or voltage required to drive the motor.

The DC servo amplifiers can be divided into two categories: linear amplifiers which need a voltage control signal, and switching (PWM) amplifiers which need a PWM control signal.

Linear servo amplifiers usually use a linear gain element such as an operational or differential amplifier to drive a power stage, the power stage then drives the motor. Because linear amplifiers control the motor voltage or current by controlling the voltage applied to the motor, there must be a voltage drop on the transistors which is equal to the difference between the supply voltage and the required motor voltage with the appropriate amount of current flowing through. Therefore, a significant amount of power is dissipated in the output transistors. The power dissipation is greatest when running the motor under low-speed high-torque conditions where the motor back emf is low and the current is high. However, the linear amplifiers are ideal when high acceleration currents are required for short time intervals where output transistor peak current ratings may be used to advantage.

Switching (PWM) amplifiers control motor voltage by varying the duty cycle of a voltage (or pulse width) applied to the motor (c.f. Fig. 4.8). The transistors operate in either an on (saturated) or off mode. Thus very little power is dissipated in either state of the transistors and the efficiency is high. Switching amplifiers are usually used in large systems, especially those which require low speed and high torque. This is the case in this research hence a switching amplifier was adopted.

4.3.2 Three Modes of PWM Drives

The basic form of PWM amplifiers is an H bridge circuit with four switching transistors and a motor schematically shown in Fig. 4.9. The four diodes are free wheeling ones to carry the "off" currents. According to their configuration, PWM amplifiers can be divided into three categories: bipolar, unipolar, and limited unipolar. The analysis method here is based that of Kuo and Tal (1978).

For convenience, let the switching frequency be f_s and switching period be t_f. Let the "on" interval occur during the first part of the period (between $t = 0$ and $t = t_1$), and let it be followed by the "off" interval (which occurs during the reminder of the switching period). The three operation modes of PWM amplifiers are then shown in Table 4.1. Detailed analysis is given by Kuo and Tal (1978).

4.4 DESIGN AND TESTING

4.4.1 Bipolar PWM Servo Drive

At beginning of this research, a bipolar PWM servo drive was adopted which is recommended by the manufacturer of the HCTL-1000 control chip (c.f. HP 1986). In the case of bipolar mode, F1 and F4 are turned on and F2 and F3 remain "off" during the "on" phase, then all FETs are turned "off" during "off" phase.

4.4.1.1 Logic design

The PWM output port in the HCTL-1000 consists of the Pulse and Sign pins and is determined by the register R09H. The PWM signal has a frequency of External Clock/100, which is 18.432KHz in this application, and the duty cycle is resolved to 1

part in 100 or 1%. The sign signal specifies the direction of rotation. The 8 bit 2's complement contents of R09H determine the duty cycle and polarity of the PWM command. Table 4.2 relates the numbers in R09H to the corresponding PWM duty cycles. The PWM Port saturates at 100% duty cycle level when the numbers in R09H are 64H to 7FH, and at -100% between 80H to 9CH respectively.

Fig. 4.10 shows the logic circuit for bipolar motor drive in this application. The function of it is to translate and/or amplify the two signals PULSE and SIGN, which are output from the PWM port of each HCTL-1000 unit on the Six Motor Controller PCB, into correct control signals to the four FETs contained in each power stage.

HCPL-2530 Dual High Speed Optocouplers contain a pair of light emitting diodes and integrated photon detectors with electrical isolation between input and output. The transfer speed is 1M bit/s and meets the requirements in the application, which is to resolve pulses of 0.5μs if a 2MHz clock is employed (it is 1.8432MHz at the moment). HCPL-2530 isolates the motor drive from the computer so as to protect the computer from damaging currents or voltages. Note that the signal ground of PULSE and SIGN is isolated from the power ground of the drive board.

U3, U4 and U5 compose logic gates to give control signal to an H bridge amplifier so that either F1 and F4 conduct or F2 and F3 conduct. This working fashion allows for bipolar motor operation with a unipolar power supply (c.f. Fig. 4.11). U5 is a ULN2823, an 8-channel Darlington Transistor Integrated Circuit, which can sink current to 500mA and withstand voltages to 90V. The application of ULN2823 is to provide sufficient gate drive to saturate the FETs, also provision is made for higher drive voltages so that higher currents can be produced to overcome the static friction of the motor brushes and the lead screw. An adjustable power supply was used for this purpose, and later replaced by a 36V traction battery.

4.4.1.2 PWM drive design

Fig. 4.11 shows the power stage of the PWM motor drive. The Power MOS (Metal-Oxide-Semiconductor) FET (Field-Effect-Transistor) offers unique characteristics and capabilities that are not available with traditional bipolar power transistors, especially when used in motor control. The most important, its switching speed is very fast, which allows efficient switching resulting in quick response, high frequency, reduced noise and small size. For example, the switching speed for the MTP15N05E power MOS FET chosen in this application (c.f. Fig. 4.12) can reach the sufficient frequency of 5MHz. Also, it has a parasitic diode between source and drain for reducing recovery time. It is forward biased when the source is at a positive potential with respect to the drain. This diode has comparatively high switching speed and the current rating is equal to that of the power MOS FET. In addition, it will allow a certain amount of current flowing through the motor then to drive the load. The MTP15N05E is an N-Channel Enhancement-Mode Silicon Gate, drain current 15A, drain-source voltage 50V power TMOS FET (c.f. Fig. 4.12).

The four 16V 4745A, 1W Zener diodes limit the gate-source input voltage of the FETs. When using a 24V power supply for the FETs and motor drive, a 36V auxiliary power supply is added, giving an extra 12V voltage to drive the gate inputs, hence ensuring sufficient input voltage for the FETs. The gate input signals are coupled through 2.2KΩ resistors.

An important noticeable point during this power drive design is the heavy current. For this reason, the arrangement of C1, C2 and R25 is for providing a discharge path when the power is turned off so as to protect the FETs and the motor. The fuse is for current over load protection. The voltage of the logic circuit is kept 5V by the 5V Zener diode and the circuit and the diode draw about 50mA current. There should be a 5V voltage drop across the relay coil, so R27 was chosen as 33Ω. The two normally

open contacts of the relay are connected to 36V power supply of ULN2823 and the 36V signal power supply. This is for protecting the bridge output stage from unexpected signal disturbances and voltages when the power is switched on. It also ensures that all power FETs are off if the logic power supply fails.

4.4.1.3 Problem and analysis

During experiments, a problem occured. The phenomenon was that when the duty cycle of the PWM signal was above 60%, the system worked; but the motor didn't run when the duty cycle was below 60%. If so, the PWM signal can only vary from 60% to 100% and the linear operation range of the motor speed is reduced, i.e., the motor would not respond to quite large errors in position or velocity. This was unacceptable.

The current and voltage of the motor drive were analyzed and measured. For a bipolar drive, the voltage of the motor V_m varied from $+V_s$ to $-V_s$ corresponding to the gate voltage change. When the motor ran under a lower PWM signal, the current decayed rapidly to zero, so the active current suppression gave a very small average current which resulted in an insufficient torque to drive the motor .

4.4.2 Limited Unipolar PWM Servo Drive

To overcome the problem of insufficient torque, a limited unipolar PWM amplifier was adopted instead of bipolar one, so that a higher average motor current could be obtained. For a limited unipolar PWM servo drive, F1 and F4 are turned on while F2 and F3 remain "off" during the "on" time. During the "off" time, F1 is turned off and F4 remains "on", with F2 and F3 "off". The logic drive was altered accordingly (c.f. Fig. 4.13).

This modification changed the motor current flow conditions (c.f. Fig.4.14). During the "on" time, V_m is equal to V_s, the current I_{on} flows through F1, the motor and F4 to GND and increases gradually. During the "off" time, F1 is switched off and F4 remains on. A current loop is formed via F4, motor and the internal parasitic diode of F2. The current starts to decay but only at a slow rate because the negative voltage across the motor is the sum of the diode and transistor drop plus the back emf of the motor. The current increases during the next "on" period so that the average current keeps high. This effect can be seen in the measured voltage and current graphs shown in Fig. 4.15. The measuring conditions are:

- The load is the gear train and when the lead screw was running against one end of the wall of the screw housing;
- 24V DC;
- 50% duty cycle.

This design proved to work well under low PWM duty cycles. During experiments, there is some transient current at the moment of the motor voltage V_m changing, this is shown in Fig. 4.16 and does not effect the operation of the motor.

4.4.3 Motor Performance

Under the limited unipolar drive mode, the motor performs well. The relationship of the motor speed versus PWM duty cycle is shown in Fig. 4.17 which was measured under voltage supply = 24V DC and zero load. Note that the operation range of the motor under the limited unipolar drive is greatly increased than its counterpart under the recommended bipolar deive. The motor can't be driven when the duty cycle <10%, because the rotor torque and shaft Coulomb friction are greater than the motor output torque. This operation range is called "dead zone". The threshold of the duty cycle, under which the motor can be driven, can be improved by increasing the voltage of the main power supply. For example, when it's 48V, the smallest duty cycle which the motor can work under is 5%.

A noticeable benefit from using the HCTL-1000 is that the PWM port has an option of Sign Reversal Inhibit for H bridge type amplifiers. If Bit 0 in the Status Register (R07H) is set, one PWM period of the pulse output is then inhibited after a sign polarity reversal. This ensures that one pair of transistors turn off before the other pair turn on and thereby avoids a short across the power supply. Fig. 4.18 shows the output of the PWM port when Bit 0 is set.

4.5 SUMMARY

After the design and testing of several drive systems, a limited unipolar PWM servo drive for permanent magnet DC motors was adopted. This provides a relatively simple and high efficiency drive system with a dead zone of less than 10%. For very large applications, a separate exited motor would be used and the motor drive could be the same type but more complex.

The PWM drive is more reliable then its linear counterpart due to the small power dissipation and the simple circuitry. For this application, a limited unipolar PWM drive can provide sufficient current to drive the motor. The change of PWM duty cycle doesn't change the waveform of the current, but the average magnitude of the current.

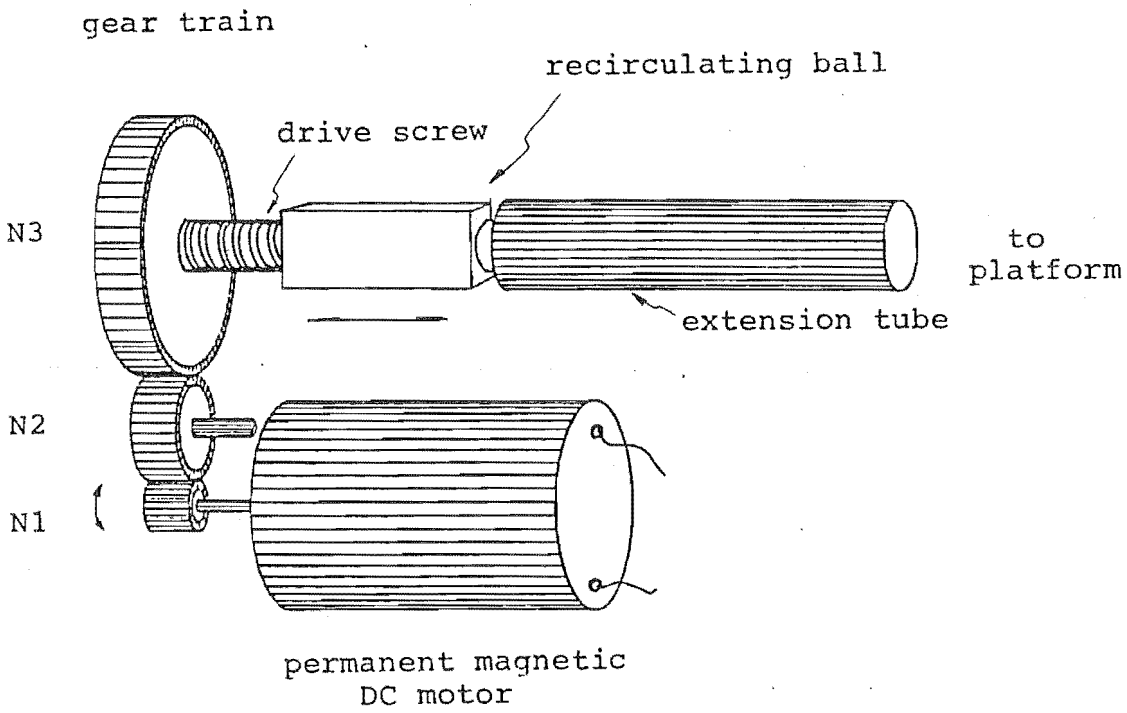


Fig. 4.1 Schematic diagram of an Electrack 100 linear actuator

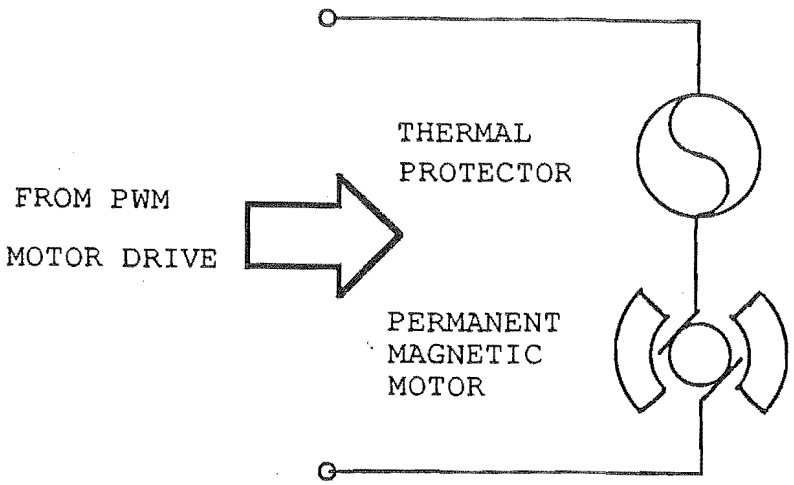


Fig. 4.2 Permanent magnet DC moto:
in the Electrak 100 actuator

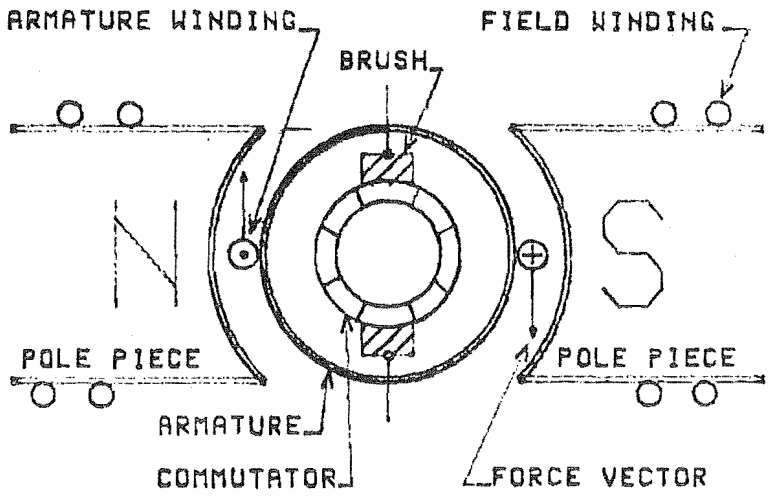


Fig. 4.3 Schematic diagram of a DC motor

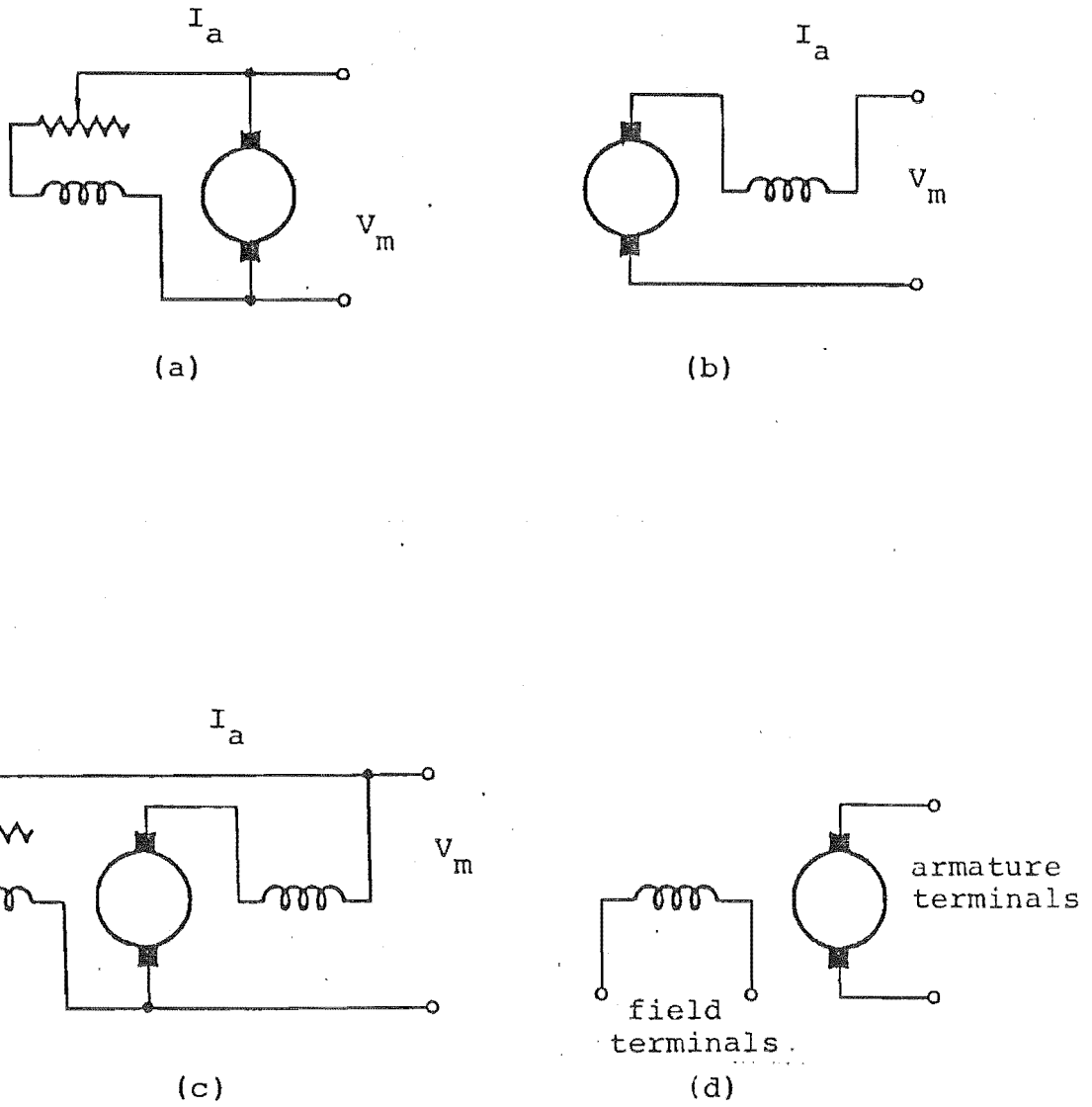


Fig. 4.4 Classifications of DC motors
(a) shunt motor; (b) series motor;
(c) compound motor; (d) seperately excited motor.

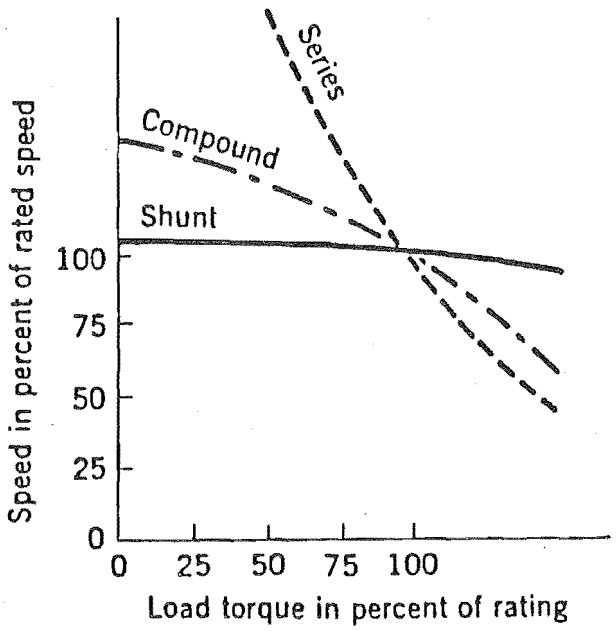


Fig. 4.5 Speed-torque characteristics of DC motors

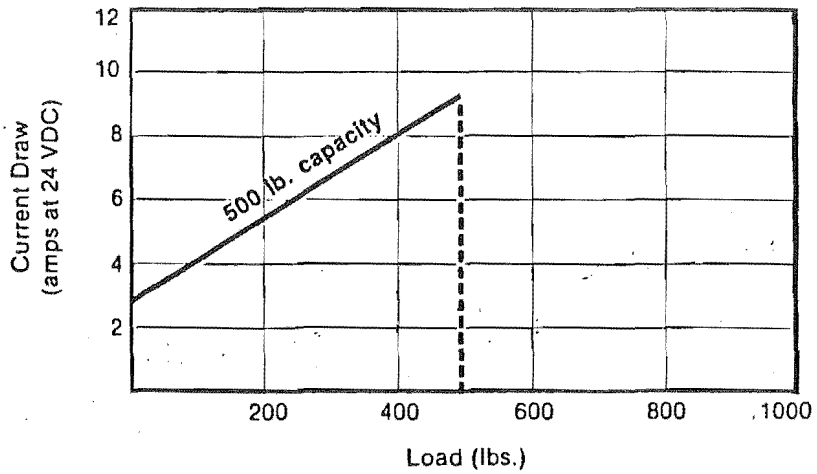


Fig.4.6(a) Load current characteristics of the Electrak 100 motor

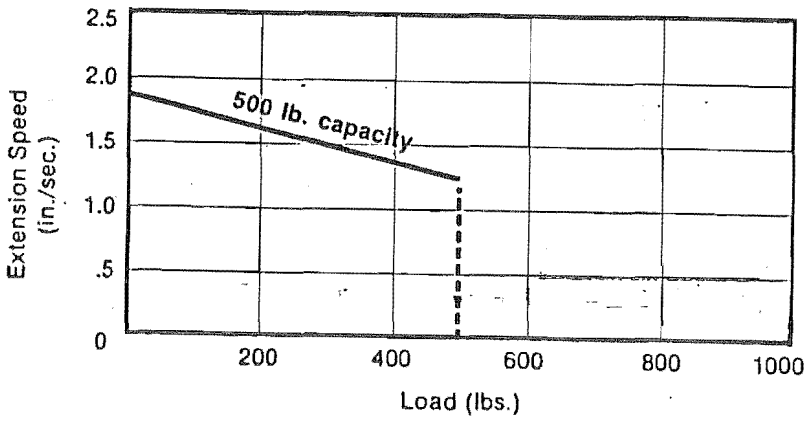


Fig. 4.6(b) Load-extension speed characteristics of the Electrak 100 motor

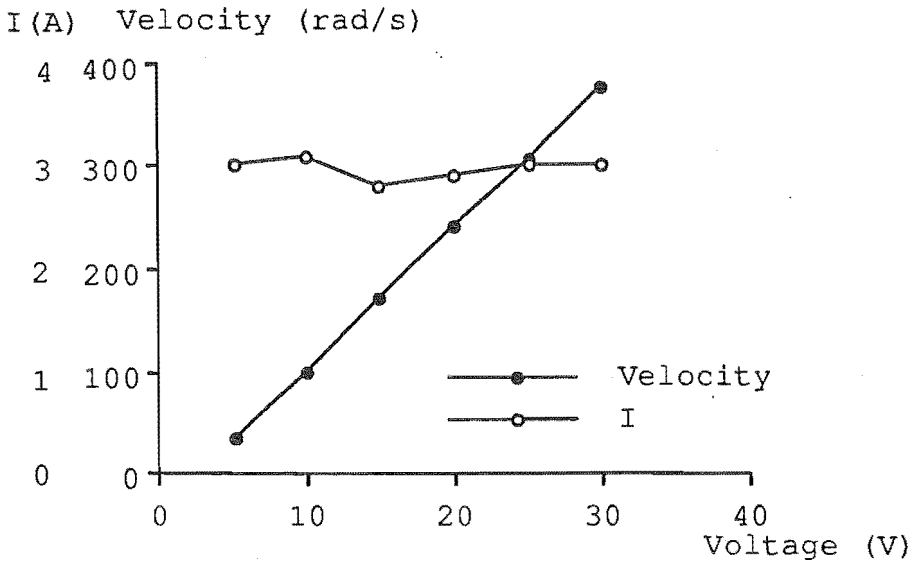


Fig. 4.7 Measured voltage,velocity and current of the Electrak 100 motor

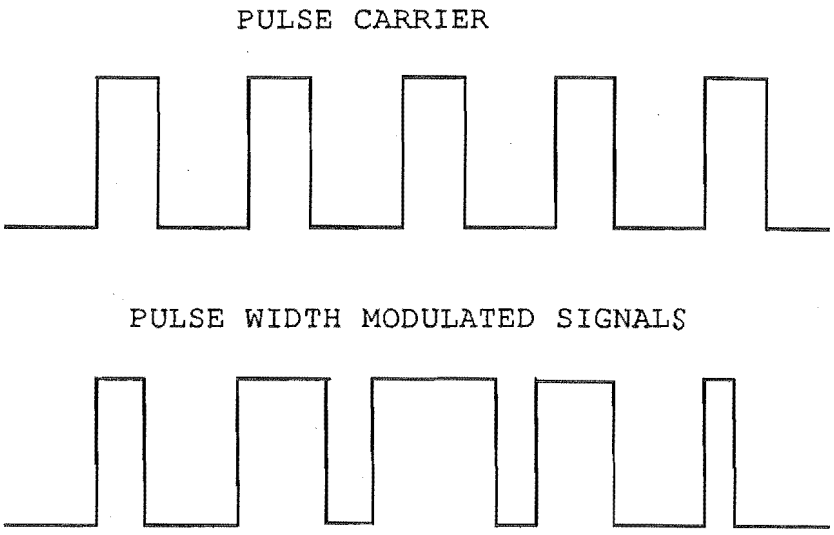


Fig. 4.8 Pulse width modulated signals

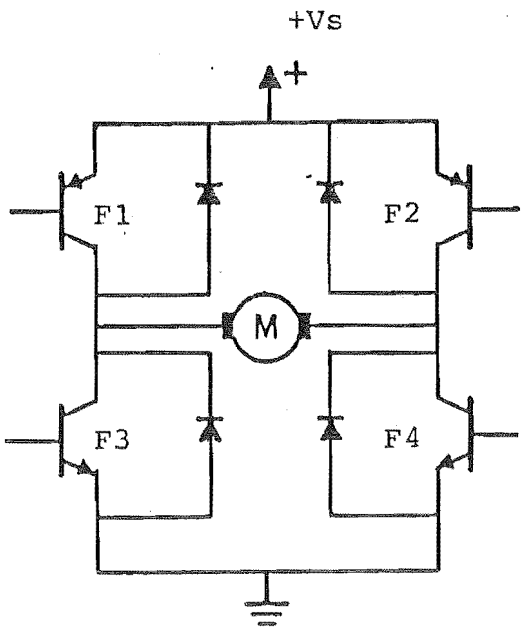


Fig. 4.9 PWM amplifier and DC motor

Table 4.1 The operation modes of PWM amplifiers

Operation Mode and Input Voltage V_{in}	Transistors Condition and Motor Voltage	
	"On" Phase	"Off" Phase
Bipolar mode	F_1, F_4 on F_2, F_3 off $V_m = V_s$	F_2, F_3 on F_1, F_4 off $V_m = -V_s$
Unipolar $V_{in} > 0$	F_1, F_4 on F_2, F_3 off $V_m = V_s$	F_2, F_4 on F_1, F_3 off $V_m = 0$
Unipolar $V_{in} < 0$	F_2, F_3 on F_1, F_4 off $V_m = -V_s$	F_2, F_4 on F_1, F_3 off $V_m = 0$
Limited unipolar $V_{in} > 0$	F_1, F_4 on F_2, F_3 off $V_m = V_s$	F_4 on F_1, F_2, F_3 off $V_m = 0$ if $I_{AB} > 0$ $V_m = V_s$ if $I_{AB} < 0$ $0 < V_m < V_s$ if $I_{AB} = 0$
Limited unipolar $V_{in} < 0$	F_2, F_3 on F_1, F_4 off $V_m = -V_s$	F_2 on F_1, F_3, F_4 off $V_m = 0$ if $I_{AB} < 0$ $V_m = -V_s$ if $I_{AB} > 0$ $-V_s < V_m < 0$ if $I_{AB} = 0$

Table 4.2 The HCTL-1000 PWM port outputs

Functional Condition During Control Modes	Internal Motor Command	PWM Port		
		R09H	Pulse Duty Cycle	Sign
Minimum Motor Command	80H	80H	100%	High
Negative Internal Motor Command Saturation	>80H	8FH	100%	High
Minimum PWM Linear Range	9CH	9CH	100%	High
Zero Motor Command	00H	00H	0%	Low
Positive Internal Motor Command Saturation	<7FH	70H	100%	Low
Maximum PWM Linear Range	64H	64H	100%	Low
Maximum Motor Command	7FH	7FH	100%	Low

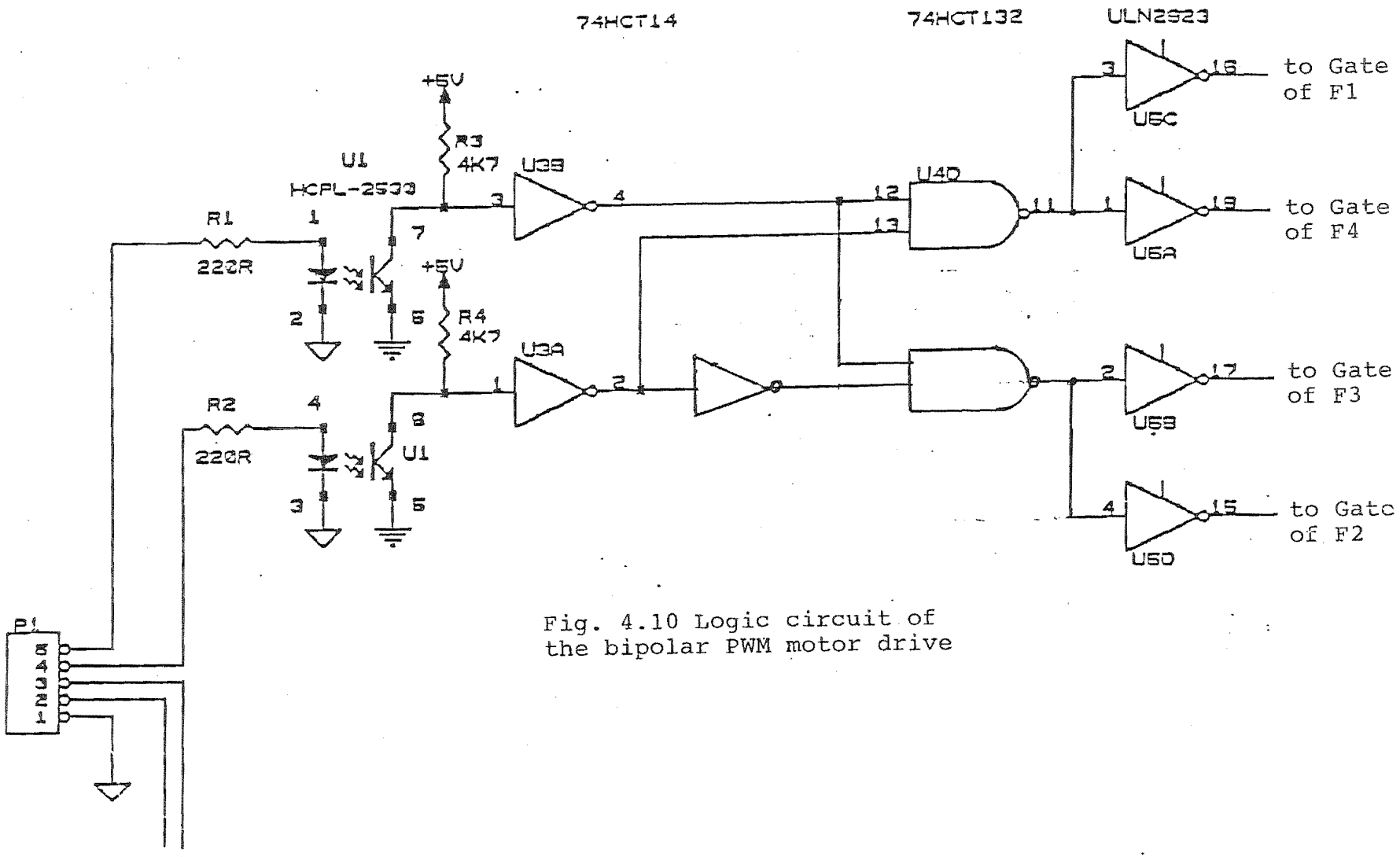


Fig. 4.10 Logic circuit of the bipolar PWM motor drive

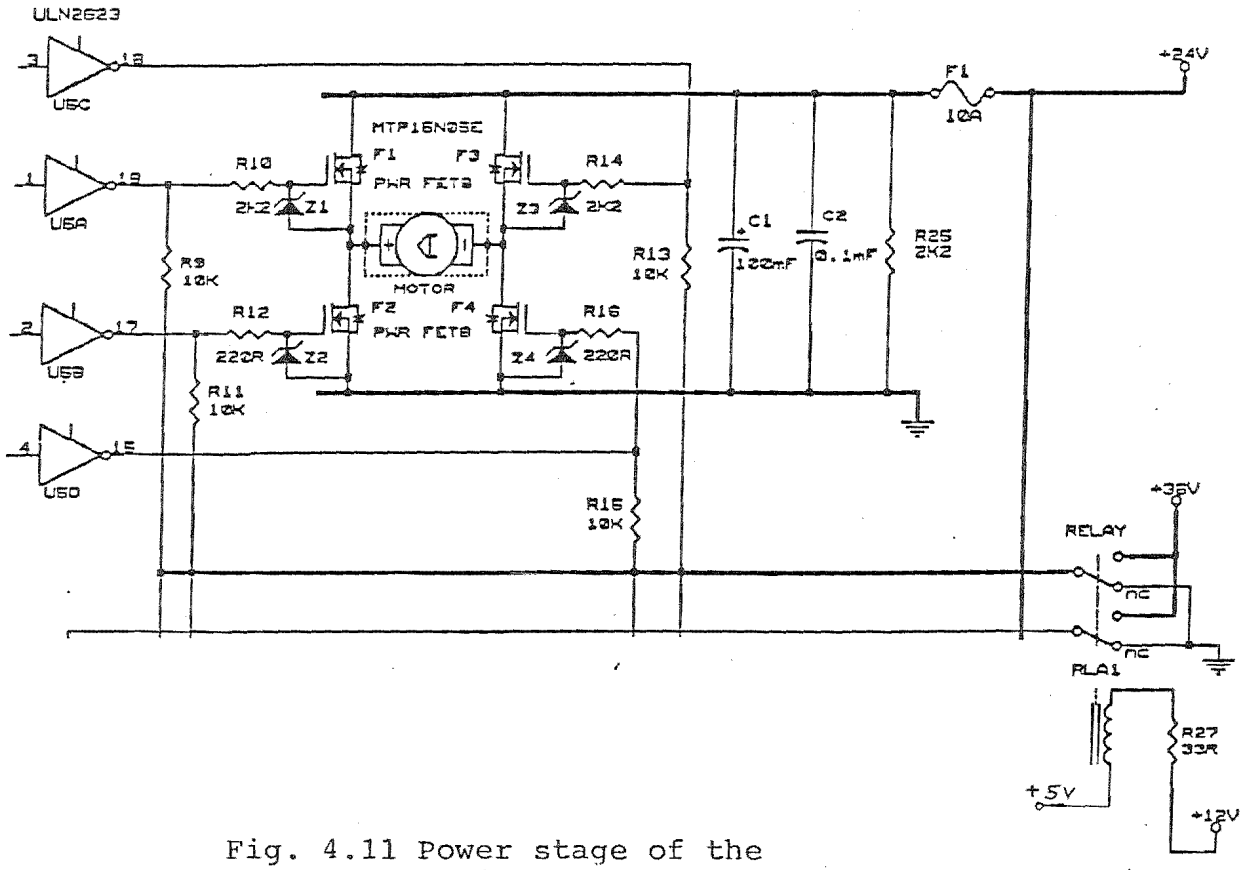


Fig. 4.11 Power stage of the motor drive

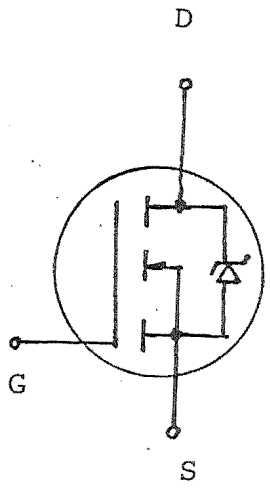


Fig. 4.12 Schematic deagram of MTP15N05E power MOS FET

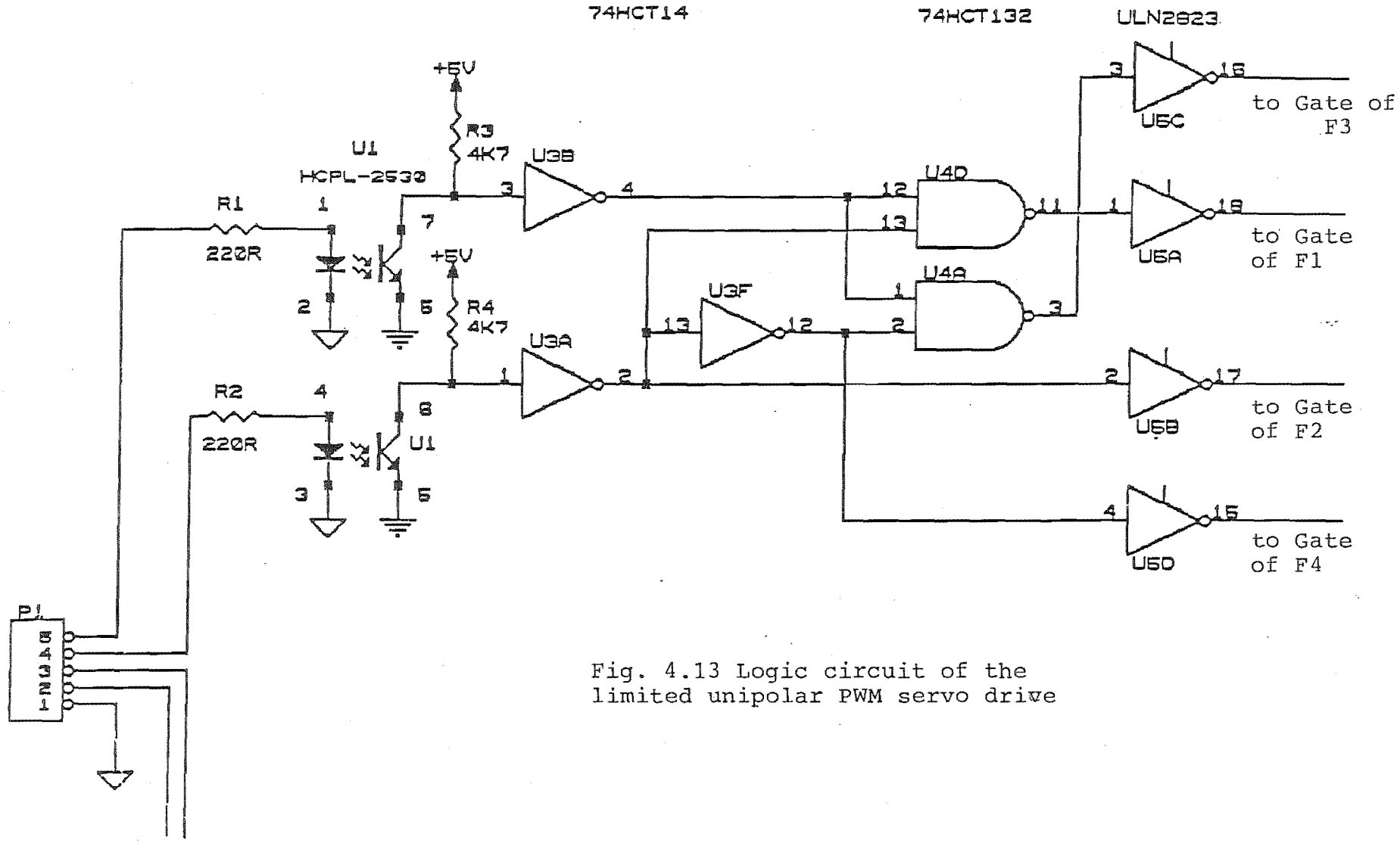


Fig. 4.13 Logic circuit of the limited unipolar PWM servo drive

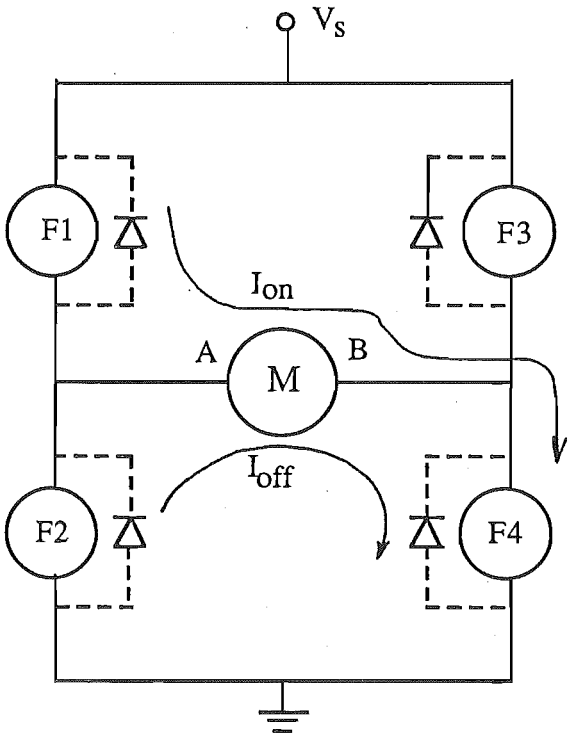
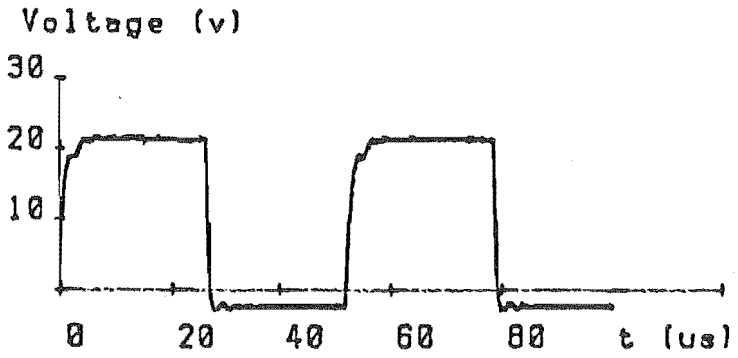
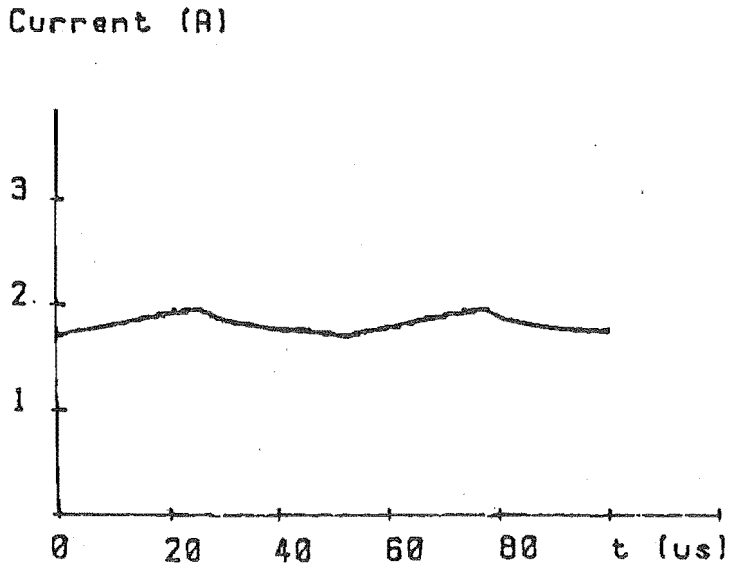


Fig. 4.14 Limited Unipolar PWM motor drive current flow



(a)



(b)

Fig. 4.15 Measured voltage and current of the limited unipolar PWM amplifier

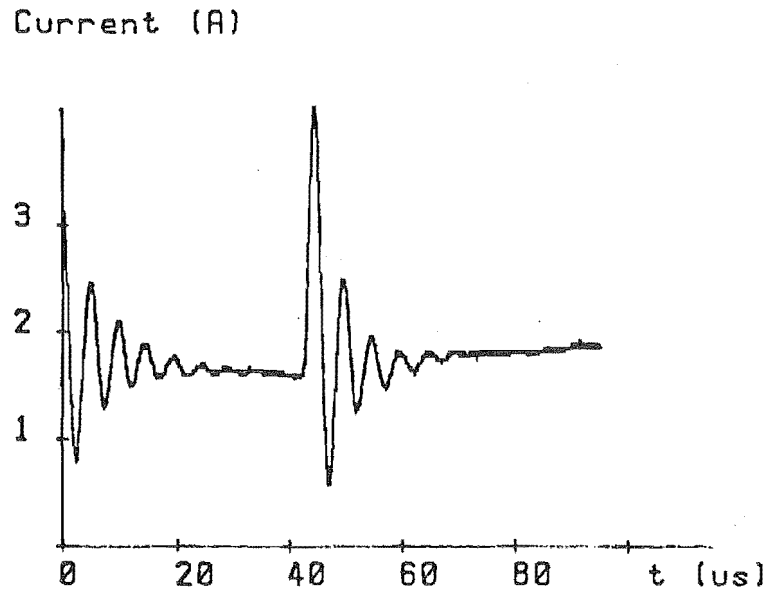


Fig. 4.16 Motor drive transient current waveforms

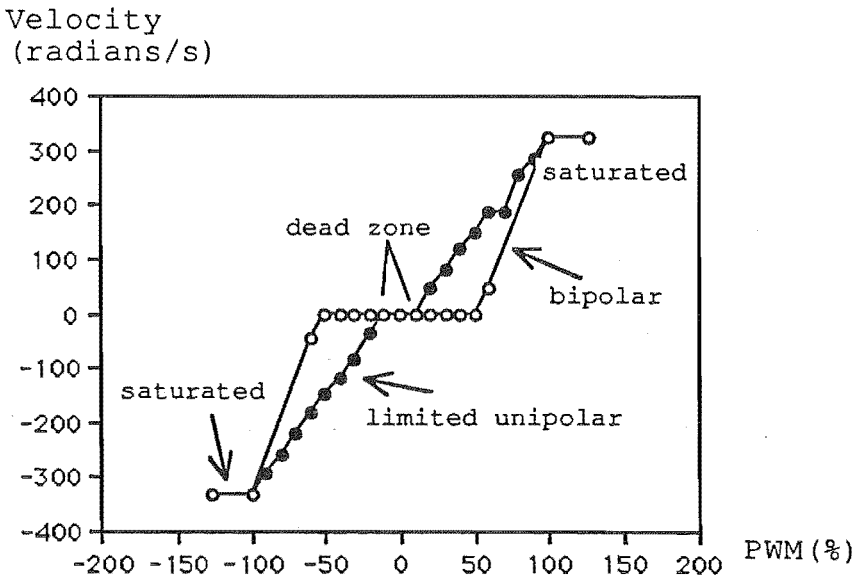


Fig. 4.17 Motor speed vs PWM duty cycle at 24V DC under limited unipolar motor drive.The PWM operation range of the bipolar drive is also shown

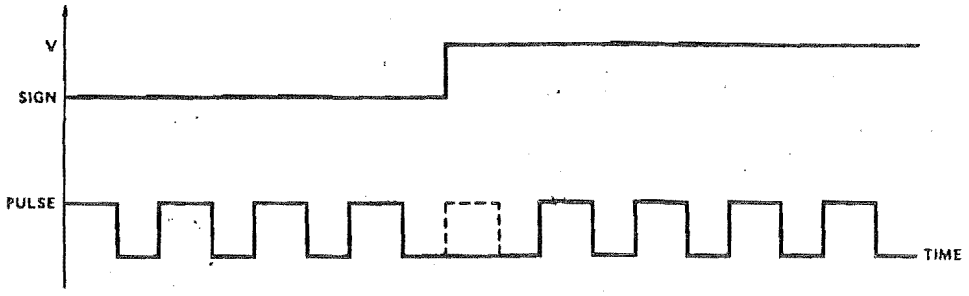


Fig. 4.18 Sign Reversal Inhibit in the HCTL-1000 (by setting Status register R07H)

CHAPTER 5

SYSTEM MODELLING AND ANALYSIS

The usual requirement for a control system is that it should have good dynamic and steady state behavior. That means the system should respond rapidly to any change in the desired specification (e.g. position or velocity control in this application) with negligible overshoot and a very small steady state error. To achieve this, control theory is employed to examine the design and to predict the performance. A mathematical model will be developed in this chapter, an analysis of the response based on the model will be studied, and finally, appropriate parameters will be selected.

5.1 CONTROL MODES

The versatile HCTL-1000 executes any one of four control modes selected by the user through software switching for different applications, which are:

Position Control;

Proportional Velocity Control;

Trapezoidal Profile Control;

Integral Velocity Control.

5.1.1 Position Control

When the Position Control mode is selected, the HCTL-1000 performs a point to point position move with no velocity profiling. A user's program is used to specify the 24

bit command position for the HCTL-1000. The required position is compared to the 24 bit actual position, which is fedback from an encoder and measured in quadrature counts, so that the position error is calculated. The position error is input to a digital filter $D_1(Z)$ which is imbedded in the HCTL-1000 and applies full digital lead compensation and outputs the motor drive command (c.f. Fig. 5.1). Once the motor reaches it's destination, the controller will remain position locked until a new position command is given.

The command and actual position data are in 2's complemen form and reside in six 8-bit registers. The order in which the registers are written or read is important. The method of combining a 3-byte 2's complement word is described in APPENDIX C. This mode provides the simplest operation among the four and hence can be conveniently used for internal algorithm testing.

5.1.2 Proportional Velocity Control

As the name indicated, the output velocity of this mode is only proportional to the input one. It performs control of motor speed using only the digital filter gain factor K for compensation. The dynamic pole and zero of lead compensation are not used. The algorithm uses the specified command velocity and the calculated actual velocity to compute the velocity error. The velocity error is multiplied by K/4 and output as motor command. The controller tracks the command velocity continuously. The system behavior after a new velocity command is governed only by the system dynamics, until a steady state velocity is reached. The units of velocity are encoder quadrature counts/sample time. The command velocity includes a fractional part for better resolution.

It has been found that the data in the unlatched Actual Velocity Registers R34H and R35H can not be read properly during movement. It is better to differentiate the actual position to get the actual velocity.

5.1.3 Integral Velocity Control

Integral Velocity Control performs continuous velocity profiling which is specified by a command velocity and command acceleration (c.f. Fig. 5.2). The user can change velocity and acceleration any time to continuously profile the velocity in time. Once the specified velocity is reached, the HCTL-1000 will maintain that velocity until a new command is specified. Going from present velocity to a new one, the present linear acceleration is followed. Internally, the controller performs velocity profiling through position control, therefore the complete dynamic compensation $D_1(Z)$ is used. From the user specified command velocity and acceleration, the controlle internally generates the required position profiles. In control theory terms, integral compensation has been added and therefore, this system has zero steady state velocity error. However, the detrimental aspect of this compensation is that, the loop stability compensation is more difficult to achieve.

This mode is used in the project so that the antenna can track satellites according to the required velocity and acceleration profile. Unexpected position changes are followed and the sophisticated coordination of six motors' movement is kept.

5.1.4 Trapezoidal Profile Control

Trapezoidal Profile Control performs point to point position moves and profiles the velocity trajectory to a trapezoid or triangle. The motion is determined by the specified acceleration, maximum velocity and final position. The HCTL-1000 computes the necessary profile to conform to the command data. If the maximum velocity is reached

before the distance halfway point, the profile will be trapezoidal, otherwise the profile will be triangular. Possible trajectories with this mode is shown in Fig. 5.3. When the move is finished, the position control locks on the final position. The controller actually performs position control while the profile generator continuously updates profile data into the Command Position registers. Full digital compensation is applied by the digital filter.

During the above four control modes, the HCTL-1000 also provides commutator control. The user can properly select the phase sequence for electronic commutation of multiphase motors like step or brushless DC motors.

A program was written to run the motor under each control modes. The flowchart is shown in Fig. 5.4.and the program is attached in APPENDIX C.

5.2 THE ELECTROMECHANICAL MODEL

5.2.1 The Motor

As indicated in last chapter, the motor drive adopted is a PWM drive in which the current is variable while the voltage is a constant. This voltage relations of permanent magnet DC motor, derived from Eqs. (4.3) and (4.5), can be expressed as

$$V_m(t) = K_e \omega + R_a I_a \qquad (5.1)$$

Here the impressed voltage is treated as the function of time t explicitly. At any time instant, the mechanically developed torque must be equal and opposite to the sum of the torques necessary to overcome the friction, inertia, and load torque. When the friction (viscous and Coulomb) torque is neglected, this developed torque is:

$$T_m = J \frac{d\theta}{dt} \tag{5.2}$$

Where, J – total moment of inertia of the rotor and the load referred to the motor shaft;

$\frac{d\theta}{dt}$ – angular acceleration.

This torque must be equal to the electrically developed torque as shown in Eq. (4.4), so,

$$K_t I_a = J \frac{d\theta}{dt} \tag{5.3}$$

Thus, the combination of Eqs. (5.1), (4.4) and (5.2) gives the relationships of voltage and angular velocity:

$$V_m(t) = \frac{R_a J}{K_t} \frac{d\theta}{dt} + K_e \omega \tag{5.4}$$

Taking Laplace transform of both sides of (5.4) leads to

$$V_m(s) = \left(\frac{R_a J}{K_t} s + K_e \right) \Omega(s) \tag{5.5}$$

So, the transfer function of the motor becomes

$$G_m(s) = \frac{\Omega(s)}{V_m(s)} = \frac{K_t}{R_a J s + K_t K_e} = \frac{K_m}{1 + \tau s} \tag{5.6}$$

Where, $K_m = \frac{1}{K_e} = \frac{1}{0.073} = 13.7$ (radian/volt.s), K_m is called the motor constant;

$\tau = \frac{R_a J}{K_t K_e}$, is called the motor time constant. By experimental measurement, the time constant $\tau = 0.114$s.

Substitute these value into (5.6), the motor transfer function is

$$G_m(s) = \frac{13.7}{0.114\ s + 1}$$

Then the angular position is:

$$\Theta(s) = \frac{K_m\ V(s)}{s\ (1 + \tau\ s)} = \frac{13.7\ V(s)}{0.114\ s^2 + s} \qquad (5.7)$$

5.2.2 The Encoder

The feedback in the system is achieved by an incremental encoder. The principle of the encoder is schematically shown in Fig. 5.5(a). The rotating disk which has 50 radial slots is mounted on the shaft of the motor. There are two OPB960 Slotted Optical Switches fixed on the motor bracket, apart from each other by almost 180°. The rotating disk is inserted through the slot of each optical switch. Each switch has a light source and a light sensor. Once the sensor receives light, the output will be a high voltage "1", else the output will be a low voltage "0". When the disk rotates, its teeth block the light passageway discretely and two trains of rectangular pulses will be output from the switches.

The phasing of the two output pulse trains, one is called channel A (CHA) and the other is called channel B (CHB), is adjusted to be 90° apart, i.e. in quadrature, as shown in Fig. 5.5(b). These signals from CHA and CHB are output to the quadrature decoder/counter in the HCTL-1000, (each interval is counted 4 times to give 200 counts per revolution) as feedback signals of the motor shaft position. The quadrature decoder/counter can detect the direction of the rotation by decoding the relative angles of CHA and CHB, and count the number of pulses in 2's complement word form.

The number then is sent to the Actual Position registers R12H, R13H and R14H to give the shaft mechanical position information. When combined with the lead screw pitch of 0.2", the resolution is 0.001" per count.

5.2.3 The Control System Modelling

An overall system diagram of the computer, the HCTL-1000 and the motor drive is shown in Fig. 5.6(a). The input, output and transfer functions of each part are indicated.

R(Z) is the Z-transform of the digital input from the host processor IBM-PC. $D_1(Z)$ is the programmable digital filter imbedded in motor controller chip HCTL-1000. The digital filter is used to compensate for closed loop system stability. The compensation is a first order lead filter in combination with the sample timer T, which is stored in register R0FH of HCTL-1000 and effects the dynamic step response and stability of the control system. It has the form

$$D_1(Z) = \frac{K}{4}\left(\frac{Z - A/256}{Z + B/256}\right) = K_g\left(\frac{Z - a}{Z + b}\right) \qquad (5.8)$$

Where, Z – digital domain operator;

K_g = K/4, K is the "gain" number in R22H;

a = A/256, a is the zero of the filter;

b = B/256, b is the pole of the filter;

A and B are manufacturer defined "zero number" and "pole number" residing in the R20H and R21H registers.

Note that the maximum output of the digital filter has been limited to 100 by saturated PWM pulses (giving 100% duty cycle) (c.f. Fig. 4.18). The error e(k) between the system input and output is

$$e(k) = r(k) - y(k) \tag{5.9}$$

This error (Z form) then is input into the digital filter $D_1(Z)$ to become G(Z). So the transfer function is

$$D_1(Z) = \frac{G(Z)}{E(Z)} = K_g\left(\frac{Z - a}{Z + b}\right)$$

The output sequence g(k) from the filter is described as

$$g(k+1) = K_g\, e(k+1) - a\, K_g\, e(k) - b\, g(k) \tag{5.10}$$

Eq. (5.10) becomes an evaluating equation for the digital filter performance. So the final output of the digital filter is

$$f(k) = \begin{cases} g(k) & g(k) < 100; \\ 100 & g(k) >= 100 \end{cases} \tag{5.11}$$

The PWM output is equivalent to a digital to analog output. This is represented in Fig. 5.5 by the impulse train generator and zero-order hold which are imbedded in HCTL-1000 unit to convert digital signals to analog signals. The zero-order hold has the form

$$\frac{1 - e^{-sT}}{s}$$

Where, T is the sample time. Then, for the part within dotted line in Fig. 5.6 (a), the analog transmittance $D_2(s)$ is

$$D_2(s) = \left(\frac{1 - e^{-sT}}{s}\right)\left(\frac{K_m}{1 + \tau s}\right)\left(\frac{1}{s}\right)$$

$$= K_m (1 - e^{-sT}) \left(\frac{1}{s^2} - \frac{\tau}{s} + \frac{\tau}{s + \frac{1}{\tau}}\right) \qquad (5.12)$$

hence the corresponding Z-transmittance is

$$D_2(Z) = K_m (1 - Z^{-1}) \left(\frac{TZ}{(Z-1)^2} - \frac{\tau Z}{Z-1} + \frac{\tau Z}{Z - e^{-T/\tau}}\right)$$

$$= K_m \frac{(nZ - p)}{(Z-1)(Z-c)} = K_1 \frac{(Z-h)}{(Z-1)(Z-c)} \qquad (5.13)$$

Where, $c = e^{-T/\tau}$;

$n = T + \tau (c - 1)$;

$p = T c + \tau (c - 1)$;

$K_1 = K_m n$;

$h = p / n$.

The transfer function

$$D_2(Z) = \frac{K_1 (Z - h)}{(Z-1)(Z-c)} = \frac{Y(Z)}{F(Z)}$$

So this discrete time system overall output sequence is described by a difference equation:

$$y(k+2) = K_1 f(k+1) - h K_1 f(k) + (1+c) y(k+1) - c y(k) \qquad (5.14)$$

Note that the third order model in Fig. 5.6(a) reduces to the second order model shown in Fig. 5.6(b) when PWM = 100%. The analysis of this model for a large step input (c.f. Fig. 5.6(b)) is given in APPENDIX A.

5.2.4 Theoretical Response

A series of initial conditions may be defined for Eqs. (5.9), (5.10), (5.11) and (5.14). Assume the input r(t) is a step input with amplitude of A_p and the output of the system $\theta(t)$ cannot respond to the input until the initial two sample time periods after. The initial conditions are

$$r(0) = A_p$$
$$r(1) = A_p$$
$$y(0) = 0$$
$$y(1) = 0$$
$$e(0) = r(0) - y(0) = A_p$$
$$e(1) = r(1) - y(1) = A_p$$
$$g(0) = K_g e(0) = K_g A_p$$

Then the theoretical response of the system can be obtained. A computer program was written to calculate the step response (c.f. APPENDIX C). One example of the system theoretical response under position control mode is shown in Fig. 5.8(a). It can be seen that in the beginning, the response is nearly linear and represents constant speed. This reflects the effect of the 100 (%) limit in the PWM signal when the error is big. This constant applied voltage is opposed by the back emf produced at constant speed. When the response approaches the specified position (A_p = 1000 quadrature counts),

the applied voltage is reduced (PWM < 100%) and the response is determined by the values of the parameters residing in the filter.

With the same parameters chosen, an experimental response of the motor was measured and plotted shown in Fig. 5.7. Comparison of the two diagrams shows that the theoretical model has a good agreement with the experimental one. Because the motor shaft friction and the armature inductance were neglected when simplifying the motor model, the slight difference in the slopes is expected, i.e., the motor's maximum speed is less than predicted when friction is neglected. The slight differences shown in the overshoot damping rate is most likely caused by Coulomb friction in the motor. This reduces the overshoot amplitudes and gives a faster damping rate. Hence, this theoretical model can be used for choosing parameters in the filter.

A detailed discussion about the effect of the friction is given in APPENDIX B. Fig. 5.8(b) is the theoretical response based on the model in APPENDIX B. Agreement with the experimental response is excellent.

5.3 EXPERIMENTAL RESPONSE ANALYSIS

Several groups of response data were collected by varying the parameters T, A, B and K in the programmable digital filter $D_1(Z)$. These were plotted in Fig. 5.9, 5.10 and 5.11.

If a closed loop system can not obtain adequate performance by using Gain K and output feedback alone, additional transmittances or compensation may be added to the system. These will improve the relative stability, steady state error or the step response overshoot.

It can be observed from Eq. (5.10) that the Gain K is most effective and depends on the last input sequence, i.e. the first to react to the input. The A has a moderate effect to compensate the K. The B plays a small part, and its effect depends on output status only. These effects can also be seen from the graphs. Sample time T has been fixed at 1111μs or R0FH = 127. Fig. 5.9 shows a group of response solely under the effect of K. When K varies from 1 to 74, it markedly changes the response from having an overdamped response with a big error to being lightly damped. In addition, the error in approaching the specified position cannot be removed only by adjusting K and not A and B as compensators. The group shown in Fig. 5.10 illustrates that the A gives an effective range of compensation to improve the stability and error when varied from 150 to 252. Although B has least effect as shown in Fig 5.11, when varied from 40 to 255, it improves the rise time slightly. It can be used as a fine compensation factor.

5.4 SUMMARY

The HCTL-1000 unit gives multiple choice in control modes which can be conveniently used in various applications to meet each particular requirement.

Except for Proportional Velocity Control mode, the algorithm of which is only governed by the Gain factor K, the other control modes all execute specified profile with lead compensation. This gives flexibility to achieve the desired motor performance.

The mathematical model of the whole system can be used as a tool for choosing the filter parameters. When the error between the input and output is small, the behavior of the system is dominated by the digital filter; and when big, by $100*D_2(Z)$ due to the saturated PWM port output.

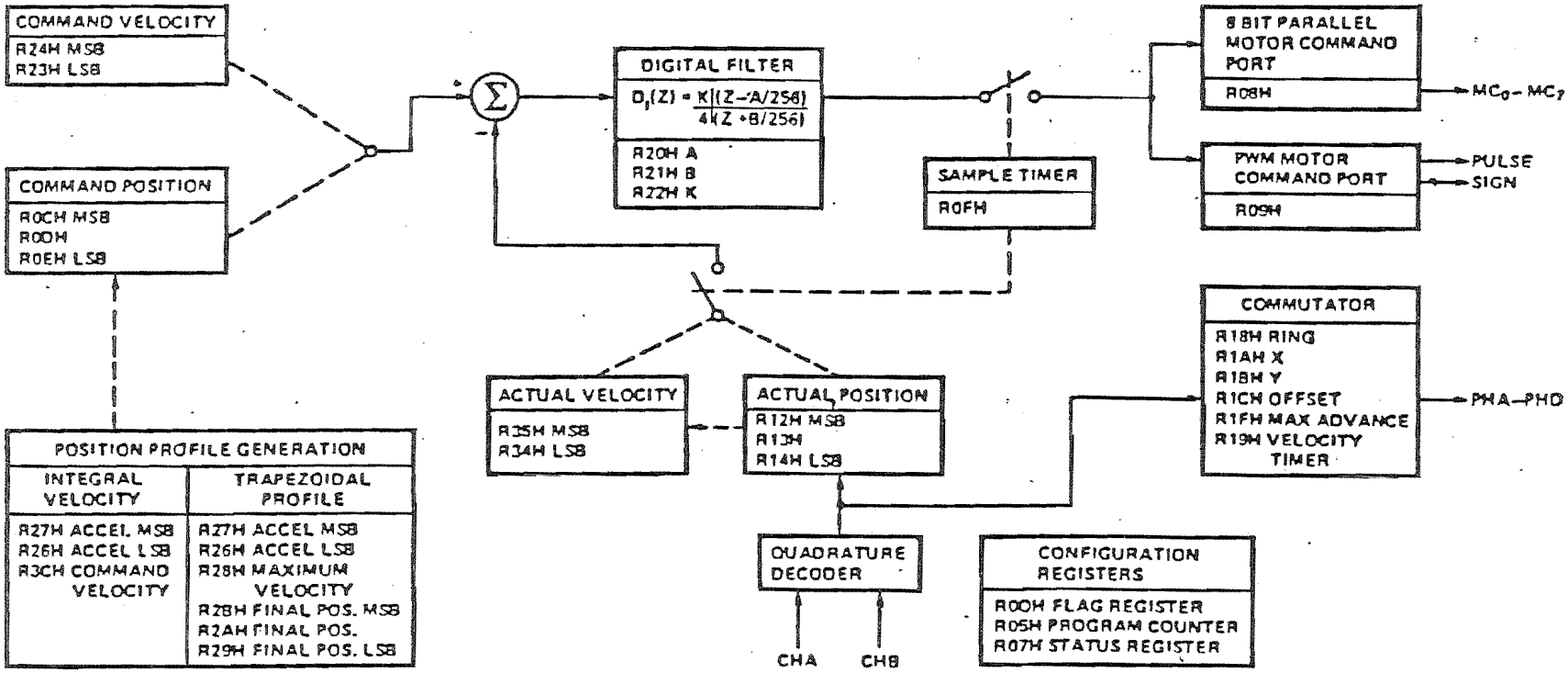


Fig. 5.1 Register Block Diagram of HCTL-1000

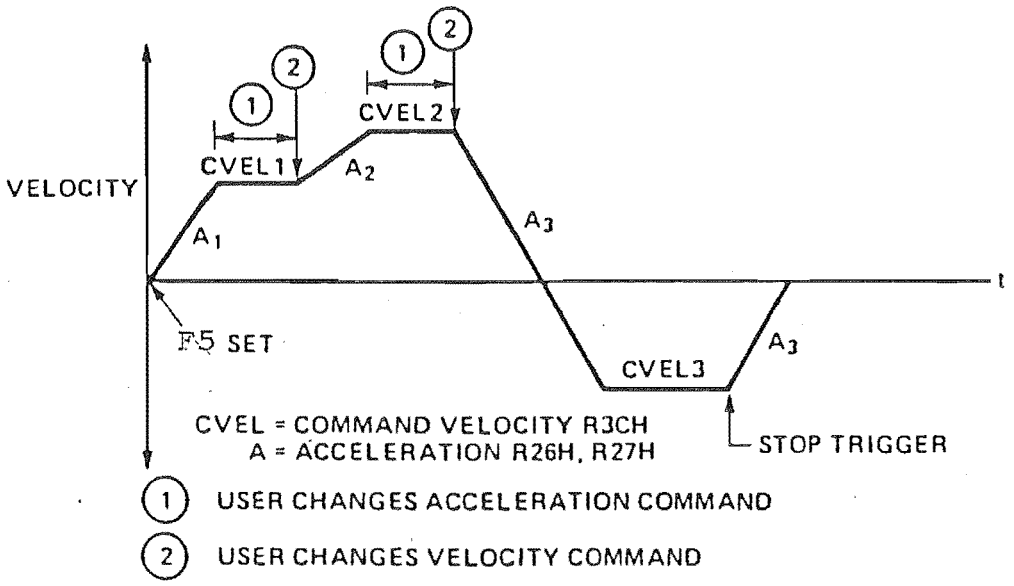


Fig. 5.2 The HCTL-1000 Integral Velocity Control Mode

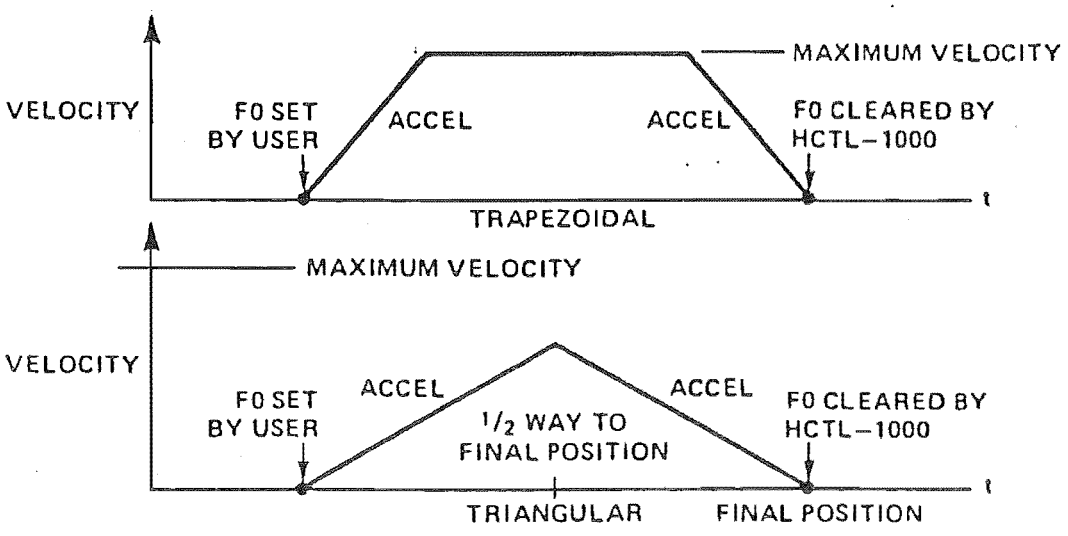


Fig. 5.3 The HCTL-1000 Trapezoidal Profile Control Mode

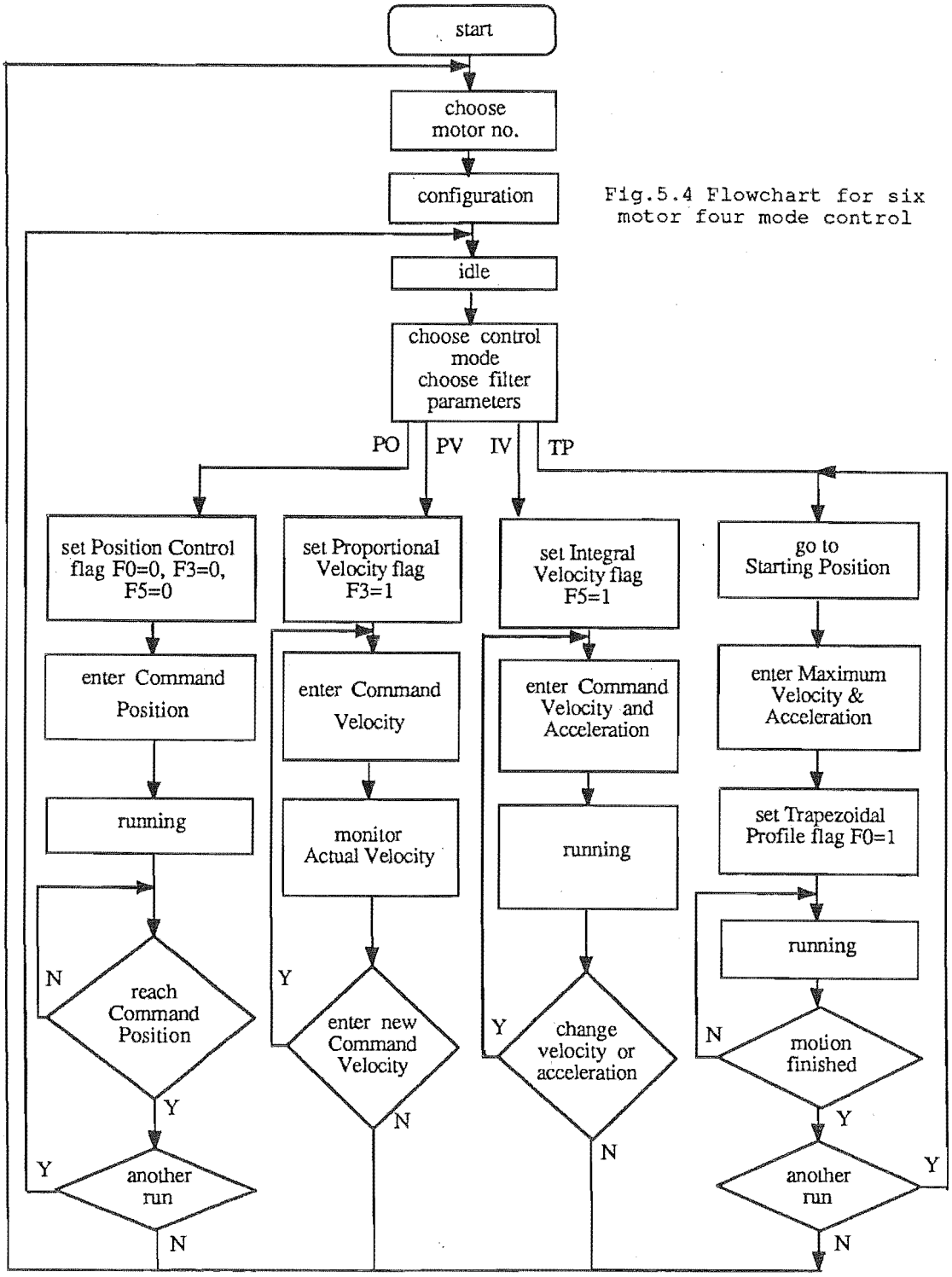
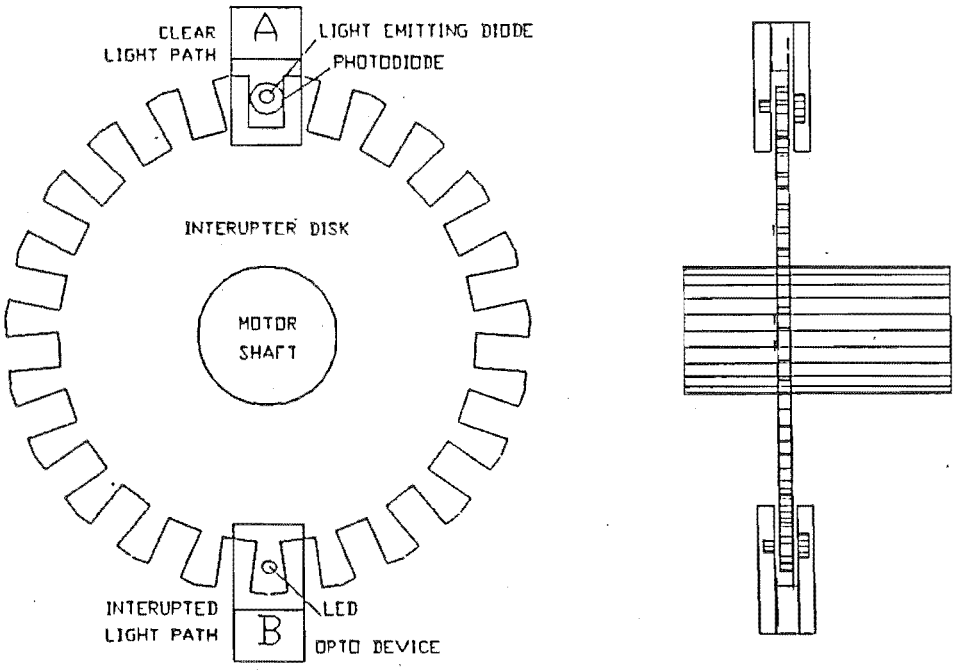
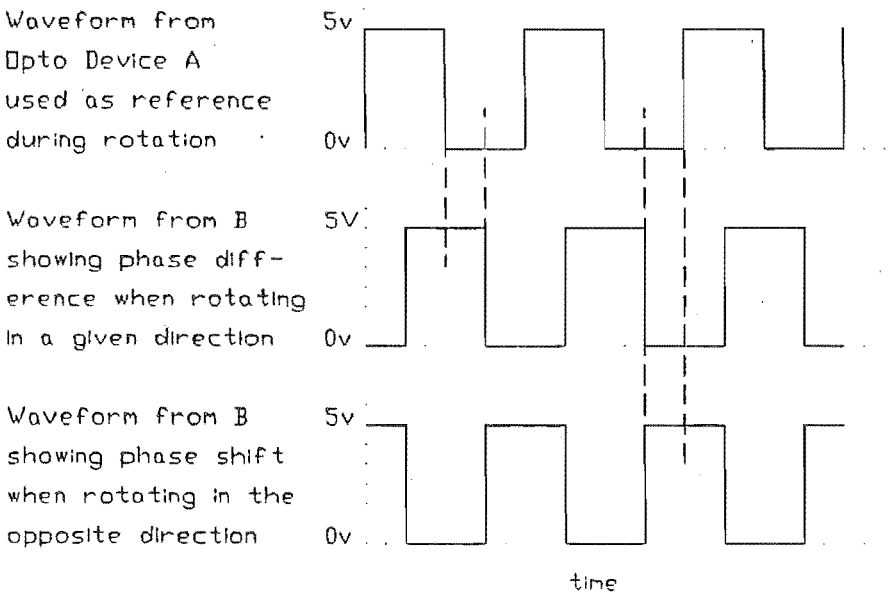


Fig.5.4 Flowchart for six motor four mode control

Fig. 5.5 Principle and output signals of the duel-channel encoder





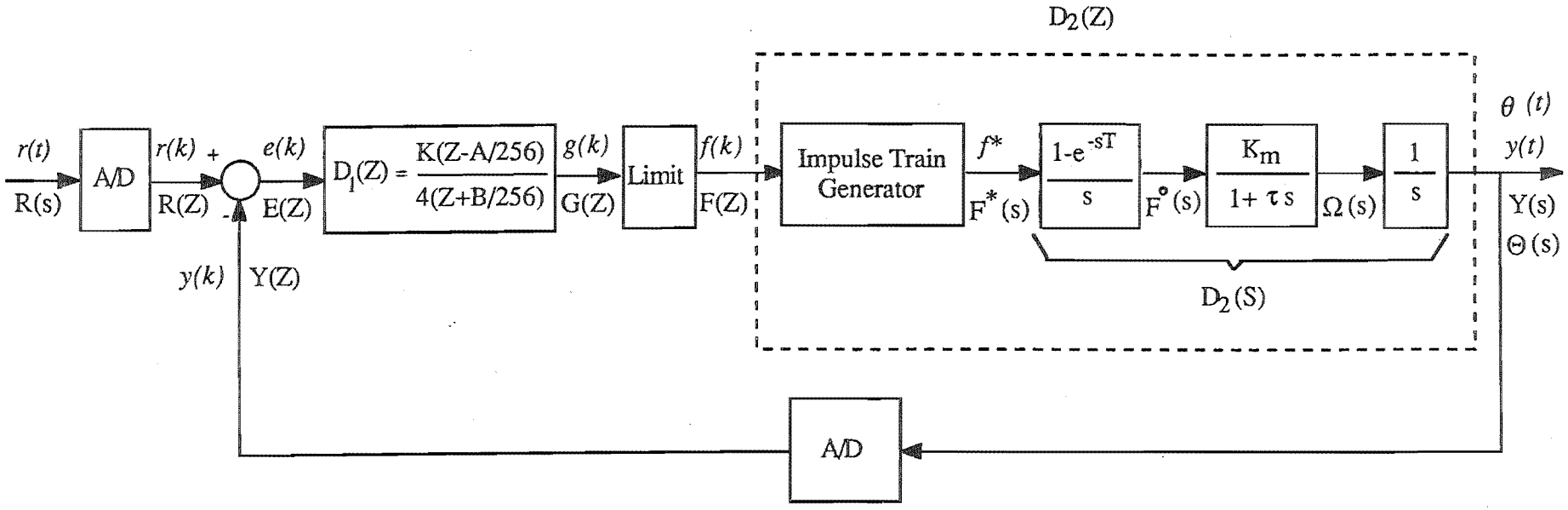


Fig. 5.6(a) Computer, HCTL-1000 and DC motor feedback control system

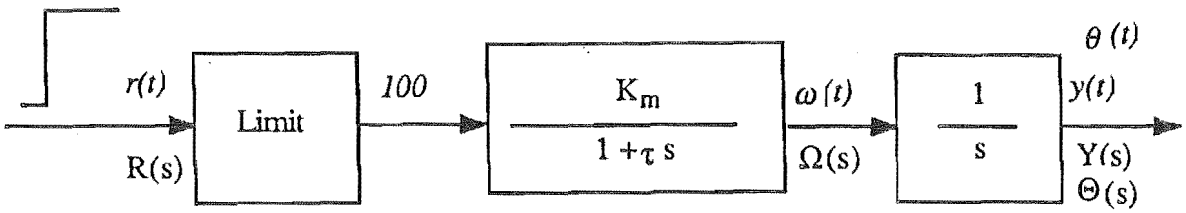


Fig. 5.6(b) System diagram under large step inputs

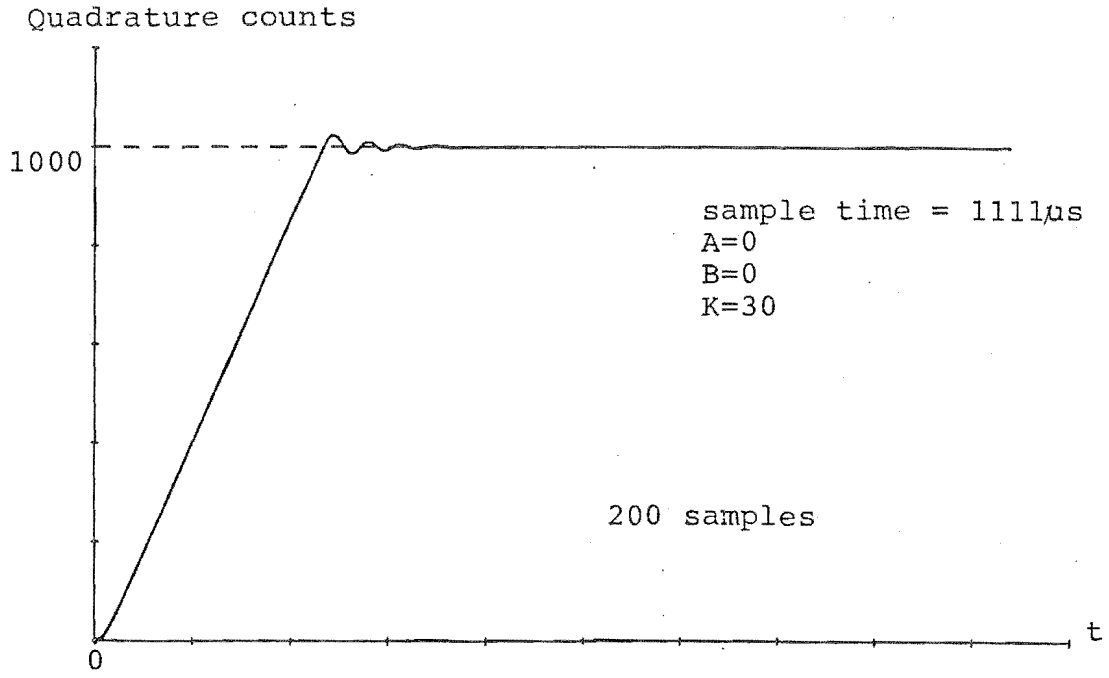


Fig. 5.7 Experimental response in position control mode (motor No.2)

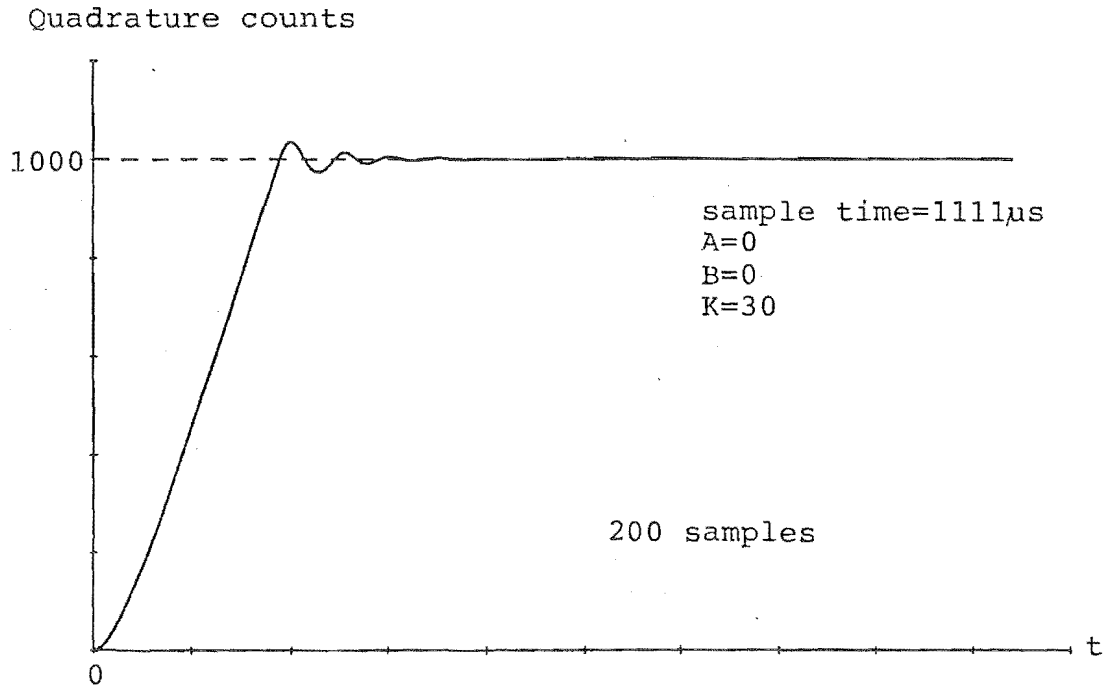


Fig. 5.8(a) Theoretical response in position control mode assuming negligible friction

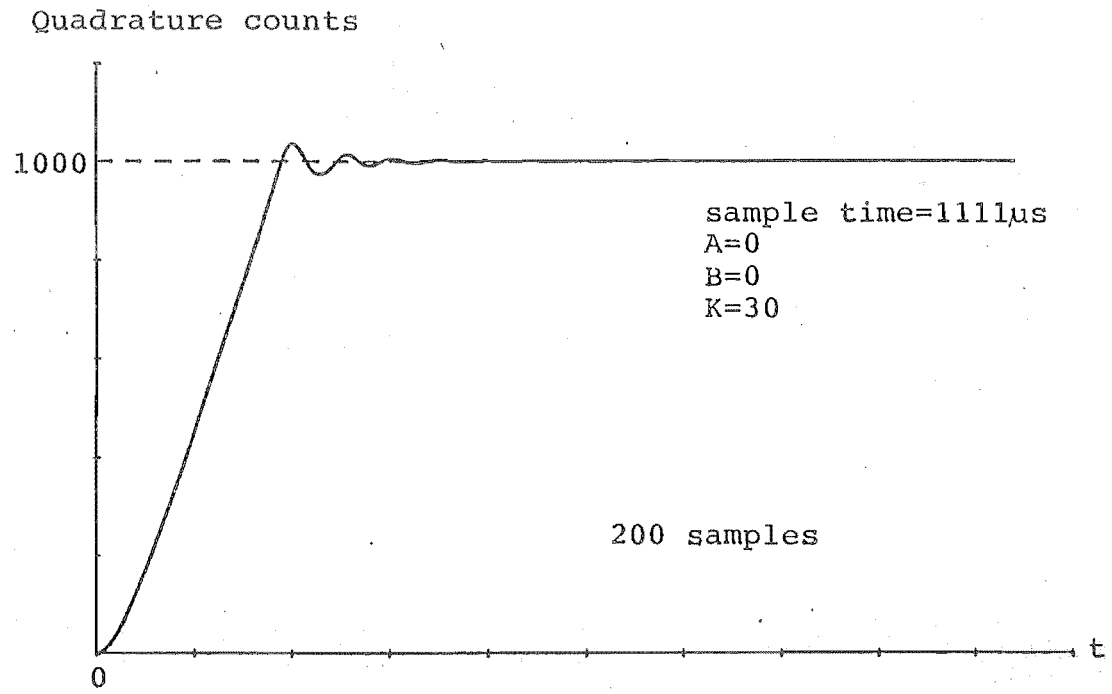


Fig. 5.8(a) Theoretical response in position control mode assuming negligible friction

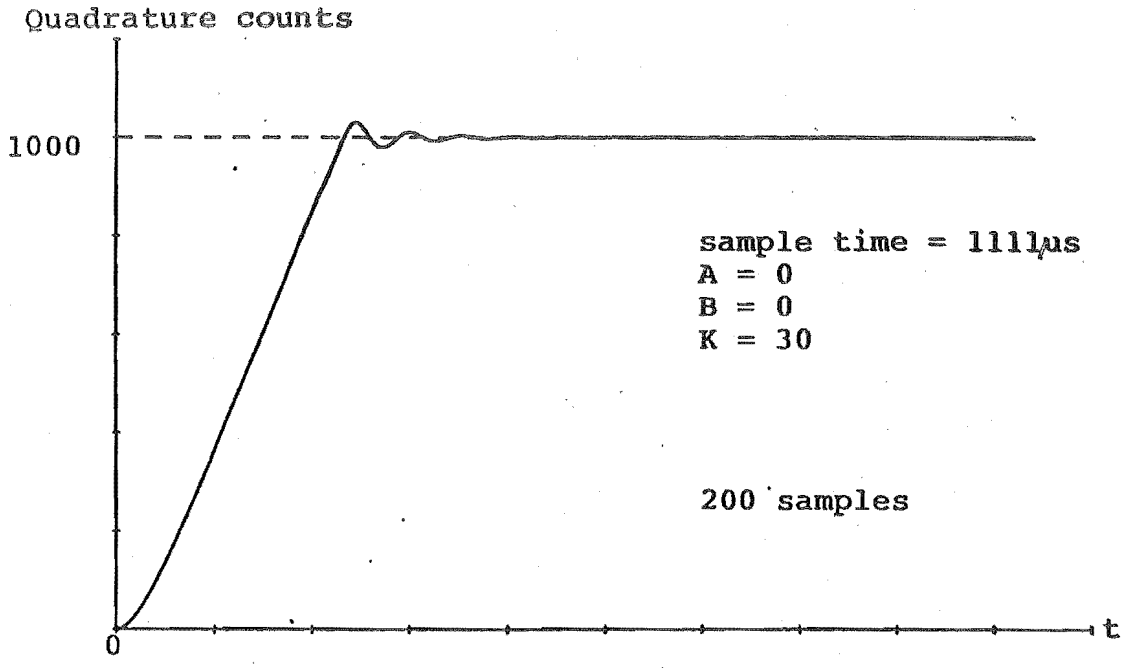


Fig. 5.8(b) theoretical response in position control mode including friction allowed for by a 20% reduction in the available supply voltage

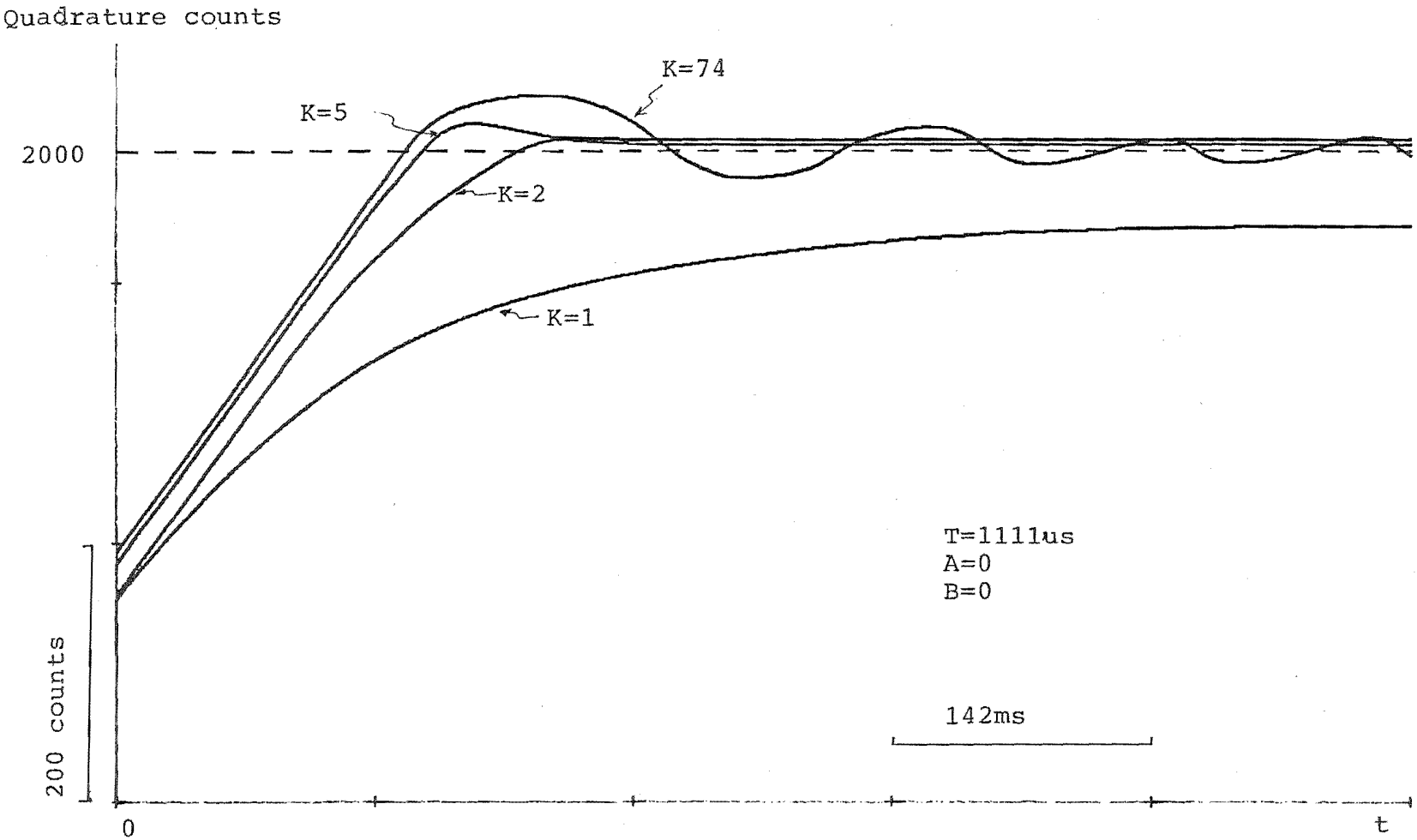


Fig. 5.9 The effect of the digital filter Gain K (from the enlarged end region of the position time profile measurement under Position Control)

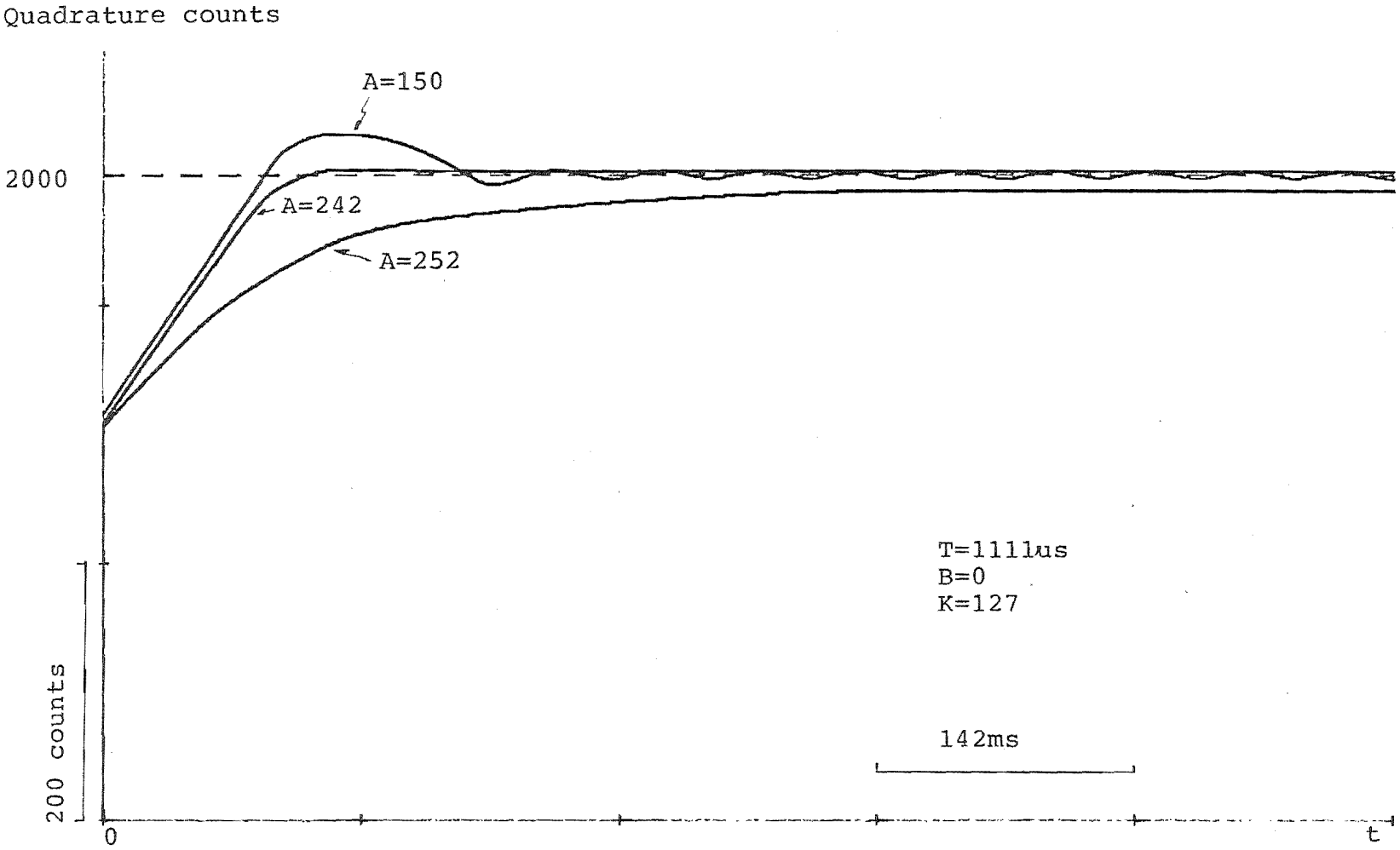


Fig. 5.10 The effect of the digital filter Zero A (from the enlarged end region of the position time profile measurement under Position Control)

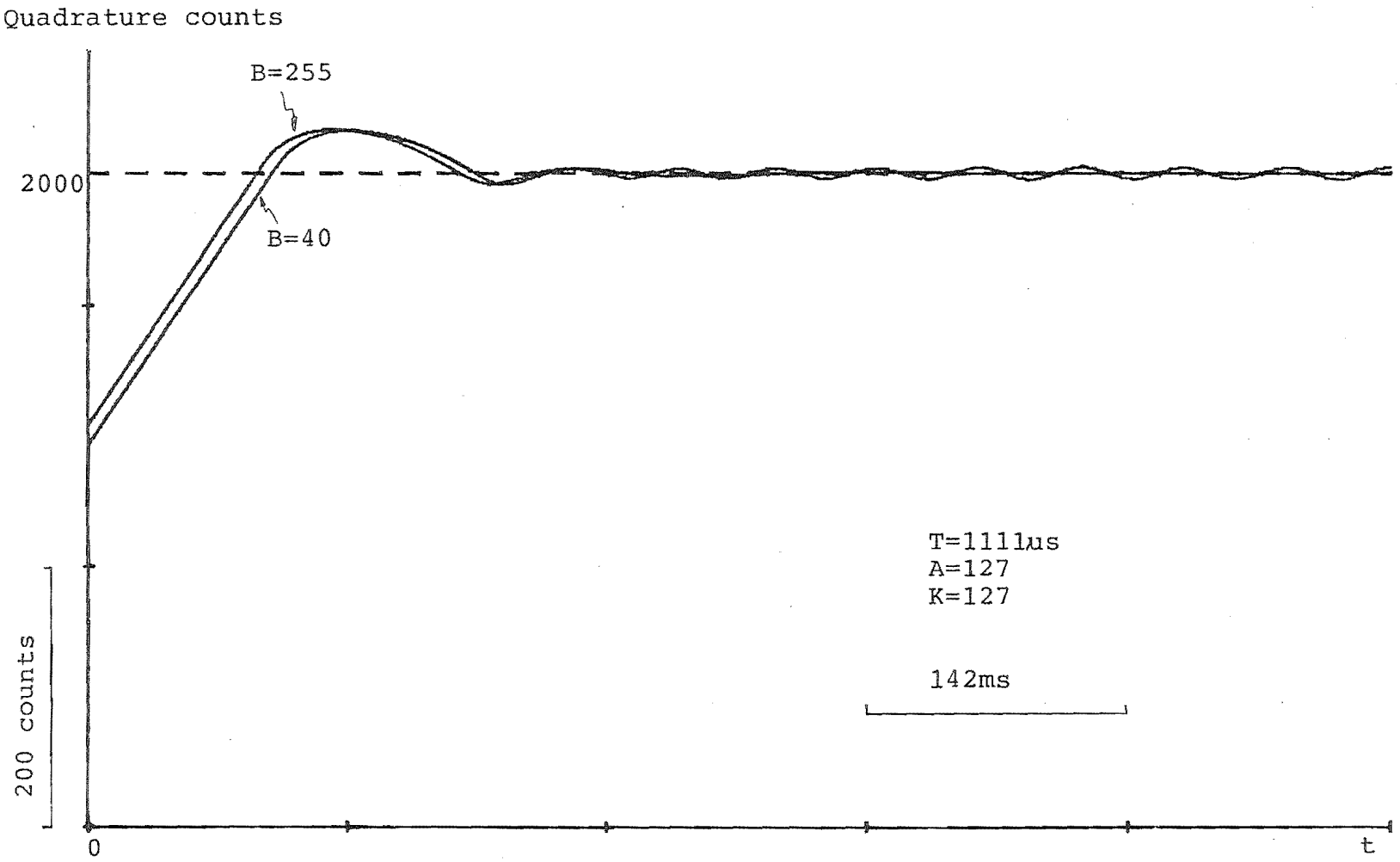


Fig. 5.11 The effect of the digital filter Pole B (from the enlarged end region of the position time profile measurement under Position Control)

CHAPTER 6

CONCLUSION

This study has embraced the analysis, design and development of multi-axis machinery numerical control.

The mechanical problems in existing satellite tracking systems and heavy duty robots and the like have been addressed by means of a multi-axis machine with six degrees of freedom i.e., a Stewart platform parallel robot mechanism.

Driving multi-axis machinery with conventional direct numerical control and a single computer software control has considerable limitations. When the speed change of the motor is high, insufficient torque will cause motor speed drop and it may stall. Hence a new type of IC controller, the HCTL-1000 has been combined with the computer to drive motors.

An IBM PC controller board based on above IC was designed to control six motors which drive the six linear actuators required for the Stewart platform. The controller has proved capable of fulfilling the fast positioning and high precision control tasks. The DC motors can be driven under Position, Proportional Velocity, Integral Velocity and Trapezoidal Profile Control modes.

In addition, a motor drive PWM servo amplifier was designed. Both bipolar and limited unipolar PWM drives were examined. The bipolar type was not suitable for this application, as it could not drive the motors when the PWM duty cycle was below

60%. The linear operating range of the motor speed control was too small. Thus a limited unipolar PWM drive stage was used. This provided sufficient current to the motor and gave a high efficiency drive system with a dead zone of less than 10% (when the rubber seals on the lead screw was well lubricated).

A mathematical model of the whole system has been developed and gave good results. When friction effects were incorporated, the theoretical model results and the experimental results are in excellent agreement. Thus the theoretical model proved useful for designing the system and predicting the system performance.

More applications can be developed from the results of this research. Drives for DC brushless and step motors, and for hydraulic cylinders can be based on the same motor controller board. These are topics for further research and development.

APPENDIX A

SYSTEM PERFORMANCE FOR LARGE STEP INPUTS

It is pointed out in 5.2.3 that, for a large step input to the system, the system performance is determined by a fixed output of PWM port and reduces to a second-order response (c.f. Eq. (5.14) and Fig. 5.7).

Assume a step input to the system is 22 volts, which is the 24V power supply less a 2 volts drop across two field effect transistors. The input is r(t) = 22 u(t) or

$$R(s) = \frac{22}{s}$$

so the output of the system is

$$Y(s) = \left(\frac{K_m}{1+\tau s}\right)\left(\frac{1}{s}\right)\left(\frac{22}{s}\right) = 22\,K_m\left(\frac{1}{s^2} - \frac{\tau}{s} + \frac{\tau}{\frac{1}{\tau}+s}\right)$$

Taking the inverse Laplace transform, the position output is

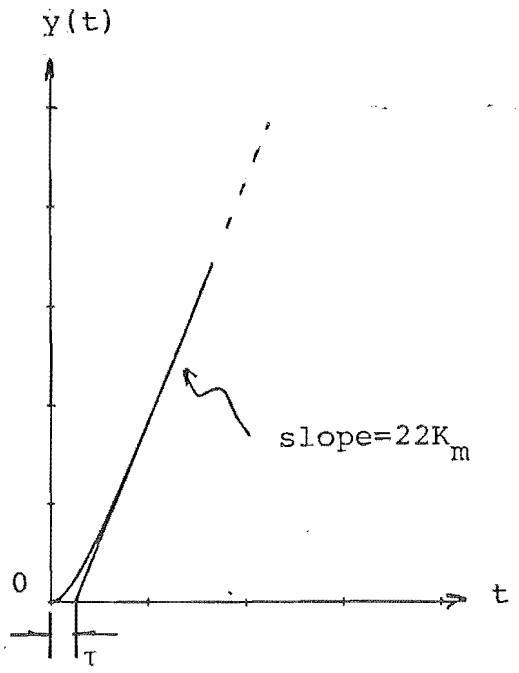
$$y(t) = 22\,K_m\,(t - \tau + \tau\,e^{-t/\tau})\,u(t) = 22\,K_m\,(t + \tau\,(e^{-t/\tau} - 1))\,u(t)$$

Therefore, the output velocity is

$$\omega(t) = 22\,K_m\,[\,1 + \tau\,(-\frac{1}{\tau}\,e^{-t/\tau})] = 22\,K_m\,(\,1 - e^{-t/\tau})$$

The second term shows the transient response after t = 0. With t increasing, the velocity tends to be constant This is observed at the beginning of the movement shown on the plotted graphs in Fig. 5.8 and the measured response in Fig. 5.7.

The theoretical response when PWM = 100% is plotted below and shows good agreement with the actual measured response.



APPENDIX B

ANALYSIS OF FRICTION EFFECTS

The effect of the friction is discussed here and a more accurate mathematical model is obtained by incorporating some simple corrections.

Current Consideration

When the model of the motor was made, the viscous and Coulomb frictions were neglected (c.f. Eqs. (5.2) to (5.6)). If the effect of the friction is considered, the extra torque T_{fr} caused by the friction must be also overcome by the developed torque T_m, and Eq. (5.3) becomes

$$K_t I_a = J \frac{d\omega}{dt} + T_{fr} \qquad \text{(B-1)}$$

From the manufacturer's graph shown in Fig. 4.6(a), the no load current is 3 Amps at maximum speed. Thus at least 3 Amps are required to overcome the friction drag. When $\frac{d\omega}{dt} = 0$ (no external load and at full steady speed), then

$$3 K_t = T_{fr}$$

So in both with load or without load conditions, Eq. (5.3) becomes

$$K_t I_a = J \frac{d\omega}{dt} + 3 K_t$$

or $$K_t (I_a - 3) = J \frac{d\omega}{dt} \qquad \text{(B-2)}$$

Note that in Eq. (B-2), I_a has an extra term "-3" which is not included in Eq. (5.3). Also note that the viscous friction force is a retarding force which is proportional to velocity. Coulomb friction is also a retarding force of constant magnitude with the sign of the frictional force changing with the reversal of the direction of velocity. In general, the friction always opposes the motor motion, so that when

$$\omega > 0,\ (I_a - 3) = (|I_a| - 3)\left(\frac{I_a}{|I_a|}\right),$$

the effective current is less hence the acceleration is reduced. In addition, the maximum speed is less than without friction. When $\omega > 0$, and $I_a < 0$, the condition occurs on position overshoot, the effective current is larger by 3 Amps. This causes the position $\theta(t)$ to settle to the specified position more quickly. In other words, the friction (mainly Coulomb friction) helps to reduce the overshoot or undershoot of the motor output. This friction effect can be observed from the experimental response graph in Fig. 5.7. The overshoot and undershoot region shows smaller overshoot amplitude and faster setting. Finally, the Eq. (5.4) becomes

$$V_m(t) = \left(\frac{R_a J}{K_t}\right)\frac{d\omega}{dt} + K_e \omega + 3 R_a$$

or,

$$(v_m(t) - 3 R_a) = \left(\frac{R_a J}{K_t}\right)\frac{d\omega}{dt} + K_e \omega$$

Voltage Consideration

The FET saturation will decrease the effective voltage by about 2 volts. The way to get around this is to recognise that the PWM signal only operates with the full voltage available. When the motor performs under the conditions of 24V power supply, 2

volts FET saturation voltage drop and the motor armature drop, the effective voltage V_{ef} is

$$V_{ef} = \alpha\ 24\ (\text{volts}) = 24 - 2 - 3\ R_a = 24 - 2 - 3 * 0.9 = 19.3\ (\text{volts})$$

So, $\alpha = 19.3 / 24 \doteq 0.80$

where α is the effective voltage coefficient. Then the Laplace transform of the motor input is

$$V_m'(s) = V_{ef}\ (\text{PWM\%}) = 24\ \alpha\ (\text{PWM\%}) = \alpha\ V_m(s)$$

where $V_m(s)$ is the original input in CHAPTER 5. So Eq. (5.6) is modified to

$$G'_m(s) = \frac{\Omega(s)}{\alpha V_m(s)}$$

or

$$G_m'(s) = \alpha\ G_m = \frac{\alpha\ K_m}{1 + \tau\ s} = \frac{K_m'}{1 + \tau\ s} \qquad \text{(B-3)}$$

where G_m' and K_m' are modified values when consider the effective voltage. K_m' reduces the gain K_1 in Eq. 5.13 by α, and thus reduces the slope of the step response.

Corrections

It has been pointed in CHAPTER 4 that the motor could be driven at 10% PWM duty cycle. Three months later when obtaining the data needed in Figs. 5.7 and 5.8, it was found that 20% PWM was needed to obtain a slow speed. Thus at least 20% of the available voltage is used simply to overcome the increased friction caused by the oiled

seals drying out. Therefore $\alpha = 0.8$ is calculated from both the voltage drop and the PWM output (i.e., 100% - 20% = 0.8).

Examine the slopes (i.e. the speeds) from Figs. 5.7 and 5.8(a), the experimental slope is

$$K_{ex} = 61.2 \text{ (radians/s)}$$

The theoretical slope (no friction allowances) is

$$K_{th} = 74.6 \text{ (radians/s)}$$

When consider the effect of α,

$$K_{th} = 74.6\ \alpha = 59.7 \text{ (radians/s)}$$

This shows a very good agreement with the experimental result. So α should be applied in the mathematical model when more accurate results are needed. When this is done, the response shown in Fig. 5.8(a) is modified to that shown in Fig. 5.8(b). The theoretical response is now very close to the measured response in Fig. 5.7.

Friction Problem

The problem noticed here is the increased friction in the motor. The extension tube support contains the rubber O-ring seals which dry out. When they are lubricated, the friction decreases. To avoid this friction problem, the seals should be replaced by grease impregnated felt seals to give smaller and more constant friction.

APPENDIX C

```
10 REM        PROGRAM1:      RNW
20 REM Reading or writion a register. The command form is RRXXDDC.
30 'RR is a LSB of a register except for R0CH (MSB for it), with
40 'an "h" end if a hex number entered. XX represents Read or
50 'Write. If a 3-byte datum to be read, enter "rrr"; if a 2-byte
60 'datum to be written, enter "ww", and so on. DD is a datum to
70 'be written to the register, with an "h" end if a hex. C is
80 'optional if reading continuously.
90 REM Press BREAK key for pause any time and press F5 for resuming.
100 REM
110 REM Configuration ********************************************
120 REM Processing input string, seperating it into three parts.
130 PRINT
140 INPUT"Enter command (all LSB but MSB for R0CH)"; X$
150 PRINT
160 A$="": B$="": C$=""
170 IF X$="" THEN 140
180 FOR P=1 TO LEN(X$)
190 Q$=MID$(X$,P,1)
200 IF Q$="R" OR Q$="r" THEN B$=B$+"R": GOSUB 310
210 IF Q$="W" OR Q$="w" THEN B$=B$+"W": GOSUB 310
220 NEXT P
230 T$=A$: GOSUB 350: R%=N#
240 T$=C$: GOSUB 350: D#=N#
250 BYTE=LEN(B$)
260 BB$=LEFT$(B$,1)
270 IF BB$="R" THEN GOSUB 410
280 IF BB$="W" THEN GOSUB 750
290 IF TT$="C" OR TT$="c" THEN 270
300 GOTO 140
310 REM seperating A$ and C$ from X$. ************************
320 IF A$="" THEN A$=LEFT$(X$,P-1)
330 C$=RIGHT$(X$,(LEN(X$)-P))
340 RETURN
350 REM Converting A$ and C$ to numbers. ***********************
360 TT$=RIGHT$(T$,1)
370 IF TT$="H" OR TT$="h" THEN T$="&H"+T$
380 N#=VAL(T$)
390 RETURN
400 REM
410 REM Reading ************************************************
420 RR%=R%
430 FOR S=1 TO BYTE
440 GOSUB 680
450 IF S=1 THEN K%=D%: K$=HEX$(D%): PRINT" LSB";D%,"or Hex ";K$
460 IF S=2 THEN J%=D%: J$=HEX$(D%): PRINT" MB ";D%,"or Hex ";J$
470 IF S=3 THEN I%=D%: I$=HEX$(D%): PRINT" MSB";D%,"or Hex ";I$
480 IF R%=20 OR R%=19 OR R%=18 THEN RR%=RR%-1 ELSE RR%=RR%+1
490 NEXT S
```

```
500 ON BYTE GOSUB 520, 580, 630
510 RETURN
520 REM Converting a 1-byte number to negitive one if it is. *
530 SUM#=K%
540 IF K%>127 THEN SUM#=SUM#-256#
550 SUM$=HEX$(SUM#)
560 PRINT"  The sum is ";SUM#,"or Hex ";SUM$
570 RETURN
580 REM Combining a 2-byte number and/or converting to negitive
590 SUM#=J%*256#+K%
600 IF J%>127 THEN SUM#=SUM#-65536#
610 PRINT"  The sum is ";SUM#
620 RETURN
630 REM Combining a 3-byte number and/or converting to negitive
640 SUM#=(I%*256+J%)*256+K%:
650 IF I%>127 THEN SUM#=SUM#-16777216#
660 PRINT"  The sum is ";SUM#: PRINT
670 RETURN
680 REM Reading a byte from a register. **********************
690 OUT &H308,RR%
700 DI%=INP(&H300)
710 D%=INP(&H308)
720 PRINT"  Read in from register ";RR%
730 RETURN
740 REM
750 REM Writing ***************************************************
760 FOR S=1 TO BYTE
770 ON BYTE GOSUB 850,890,940
780 IF S=1 THEN D%=K%
790 IF S=2 THEN D%=J%
800 IF S=3 THEN D%=I%
810 GOSUB 1000
820 IF R%=20 OR R%=19 OR R%=18 THEN R%=R%-1 ELSE R%=R%+1
830 NEXT S
840 RETURN
850 REM Converting to 2's com of a 1-byte number. ************
860 IF D#<0 THEN D#=D#+256
870 K%=D#
880 RETURN
890 REM Converting to 2's com, seperating into 2 bytes. ******
900 IF D#<0 THEN D#=D#+65536#
910 J%=FIX(D#/256)
920 K%=D#-J%*256
930 RETURN
940 REM Converting to 2's com and seperating into 3 bytes. ***
950 IF D#<O THEN D#=D#+16777216#
960 I%=FIX(D#/65536#)
970 J%=FIX((D#-I%*65536#)/256)
980 K%=D#-I%*65536#-J%*256#
990 RETURN
1000 REM Writing 1 byte to a register. ************************
1010 OUT &H308,R%
1020 OUT &H300,D%
1030 PRINT"  Output the datum";D%;"to the register";R%
1040 RETURN
```

```
REM          PROGRAM2:     PERFORM6
REM This is a general testing program for performing every
REM control mode of HCTL-1000 controller and driving six
REM motors.

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'                    CHOOSE MOTORS                                      '
'                                                                       '
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Port1=&H300                                    'set I/O addresse
Port2=&H308

Motor:
CLS
INPUT "Run which motor (Enter motor number 1, 2...)" M%
Port1=Port1+M%-1
Port2=Port2+M%-1

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'                    CONFIGURATION                                      '
'                                                                       '
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

R%=&H5: D%=0: GOSUB WriteOut                   'software reset
R%=&H7: D%=1: GOSUB WriteOut                   'PWM Sign Reversal
                                               'Inhabit

Idle:                                          'Initialization Idle
R%=&H5: D%=1: GOSUB WriteOut

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'       CHOOSE CONTROL MODE AND FILTER PARAMETERS                       '
'                                                                       '
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Order:                                         'mode and filter
                                               'command

CLS
PRINT"Enter command ss,zz,pp,gg,mm"
PRINT
PRINT"  ss: Sample Time"
PRINT"      8.6806-2222.2336us for OP and PV modes"
PRINT"      69.4448-2222.2336us for IV and TP modes"
PRINT
PRINT"  zz: Zero A, 0<=A<=255"
PRINT
PRINT"  pp: Pole B, 0<=B<=255"
PRINT
PRINT"  gg: Gain K, 0<=K<=255"
PRINT
PRINT"  mm: control mode"
PRINT"      PO for Position Control"
PRINT"      PV for Proportional Velocity Control"
PRINT"      IV for Integral Velocity Control"
PRINT"      TP for Trapezoidal Profile Control"
PRINT

'get command string and seperate it
```

```
LINE INPUT Order$
PRINT
Pie%=1: PieBg%=1
FOR Comma%=1 TO LEN (Order$)
Cma$=MID$ (Order$, Comma%, 1)
  IF Cma$="," THEN Pie$=MID$ (Order$, PieBg%, Comma%-Piebg%):_
                   PieBg%=Comma%+1:_
                   ON Pie% GOSUB SReg, ZReg, PReg, GReg:_
                   Pie%=Pie%+1
NEXT Comma%
Mode$=RIGHT$(Order$,2)

SELECT CASE Mode$
  CASE "PO", "po"
    GOTO PosiCtr
  CASE "PV", "pv"
    GOTO ProporCtr
  CASE "IV", "iv"
    GOTO InteCtr
  CASE "TP", "tp"
    GOTO TrapeCtr
  CASE ELSE
    GOTO Order
END SELECT

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'                    POSITION CONTROL                                   '
'                                                                       '
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

PosiCtr:
'set flag register
  R%=&H0: D%=0: GOSUB WriteOut
          D%=3: GOSUB WriteOut
          D%=5: GOSUB WriteOut


'tell actual position
  GOSUB AcPosi
  AcPosi# = Sum#
  CLS
  PRINT
  PRINT"The Actual Position is "; AcPosi#

INPUT"Enter Command Position -8388608<=P<=8388608";CmPosi#
'seperate input number into three bytes
  TriBy# = CmPosi#
  GOSUB TriSepe

'send into Command Position registers
  R%=&HC: D%=I%: GOSUB WriteOut
  R%=&HD: D%=J%: GOSUB WriteOut
  R%=&HE: D%=K%: GOSUB WriteOut

'ready? go!
  R%=&H5: D%=&H3: GOSUB WriteOut

MoniPosi:
  GOSUB Emerg
```

```
'monitor excution to see if it's completed
  GOSUB AcPosi: print sum#

    IF CmPosi# > AcPosi# THEN IF(CmPosi#-Sum#)>0 GOTO MoniPosi_
       ELSE GOTO Comple
    IF (CmPosi#-Sum#)<=0 THEN GOTO MoniPosi_
       ELSE GOTO Comple

Comple:
  PRINT
  PRINT"Motion completed."

  R%=&h9: D%=0: GOSUB WriteOut

  PRINT"Enter new position? (Y/N)"
  GOSUB WKey
  IF AKey$="Y" OR AKey$="y" THEN GOTO PosiCtr

  PRINT"Run another motor? (Y/N)"
  GOSUB WKey
  IF AKey$="Y" OR AKey$="y" THEN GOTO Motor

  GOTO Idle

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'             PROPORTIONAL VELOCITY CONTROL                               '
'                                                                         '
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

ProporCtr:
'set flag register
  R%=&H0: D%=11: GOSUB WriteOut

Restart:
  CLS
  INPUT"Enter Command Velocity -2047.0000<V<2047.0000";CmVel#

'deal with Command Velocity for R24H and R23H
  CV#=CmVel#*16
  IF CV#<0 THEN CV#=CV#+65536#
  J%=FIX(CV#/256#)
  K%=FIX(CV#-J%*256#)

  R%=&H24: D%=J%: GOSUB WriteOut
  R%=&H23: D%=K%: GOSUB WriteOut

'ready? go!
  R%=&H5: D%=&H3: GOSUB WriteOut

AcVel:
'read in and print Actual Velocity
  R%=&H35: GOSUB ReadIn: J%=D%
  R%=&H34: GOSUB ReadIn: K%=D%
  GOSUB TwoCombine
  PRINT" The Actual Velocity is" TAB(26) TwoSum#
  PRINT TAB(45) "The command Velocity is"; CmVel#

  LOCATE 25,1
  PRINT"Press V to give a new Command Velocity,";
  PRINT" other keys to exit"
```

```
  GOSUB Emerg

  WHILE NOT INSTAT
  GOTO AcVel
  WEND

  AKey$ = INKEY$
  IF AKey$="V" OR AKey$="v" THEN Restart

  R%=&H24: D%=0: GOSUB WriteOut          'stop motor
  R%=&H23: D%=0: GOSUB WriteOut

  PRINT"Run another motor (Y/N)"
  GOSUB WKey
  IF AKey$="Y" OR AKey$="y" THEN GOTO Motor

  GOTO Idle

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'               INTEGRAL VELOCITY CONTROL                                  '
'                                                                          '
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

InteCtr:
'set flag register
  R%=&H0: D%=13: GOSUB WriteOut

  CLS
  PRINT
  INPUT"Enter Command Velocity -127<=V<=127"; CmVel2%
  GOSUB CmVel2
  INPUT"Enter Acceleration 0.00000000<A<=127.00000000"; Acce#
  GOSUB Accele

  LOCATE 23,1
  PRINT"Enter Vnumber to give a new Velocity"
  PRINT"      Anumber to give a new Acceleration,"
  PRINT"      others to exit."

  R%=&H5: D%=3: GOSUB WriteOut

NewCom:
  GOSUB Emerg

  WHILE NOT INSTAT: WEND
  LINE INPUT X$

  VorA$ = LEFT$ (X$,1)

  Num$ = RIGHT$ (X$, (LEN(X$)-1))

    IF VorA$="V" OR VorA$="v"_
      THEN CmVel2%=VAL(Num$): GOSUB CmVel2: GOTO NewCom
    IF VorA$="A" OR VorA$="a"_
      THEN Acce#=VAL(Num$): GOSUB Accele: GOTO NewCom

  CmVel2%=0: GOSUB CmVel2                'stop motor

  PRINT"Run another motor (Y/N)"
```

```
  GOSUB WKey
  IF AKey$="Y" OR AKey$="y" THEN GOTO Motor

  GOTO Idle

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'            TRAPEZOIDAL PROFILE CONTROL                                       '
'                                                                              '
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

TrapeCtr:
'tell Actual Position
  CLS
  GOSUB AcPosi
  PRINT
  PRINT"The Starting Position is"; Sum#
  PRINT"Are you satisfied? (Y/N)"
  GOSUB WKey
  IF AKey$="Y" OR AKey$="y" THEN Start#=Sum#: GOTO FinalPosi

'utilize Position control mode
  R%=&H0: D%=0: GOSUB WriteOut
          D%=3: GOSUB WriteOut
          D%=5: GOSUB WriteOut

  PRINT"So what's your Starting Poisition";
  INPUT" (-8388608<=Start<8388608)"; Start#
'adjust Starting Position
  TriBy# = Start#
  GOSUB TriSepe

  R%=&HC: D%=I%: GOSUB WriteOut
  R%=&HD: D%=J%: GOSUB WriteOut
  R%=&HE: D%=K%: GOSUB WriteOut

'ready? go!
  R%=&h5: D%=3: GOSUB WriteOut

  GOSUB Emerg

FinalPosi:
  INPUT"Enter final Position -8388608<=Final<8388608";Final#

  TriBy#=Final#
  GOSUB TriSepe

'send out to Final Position
  R%=&H29: D%=K%: GOSUB WriteOut
  R%=&H2A: D%=J%: GOSUB WriteOut
  R%=&H2B: D%=I%: GOSUB WriteOut

  INPUT"Enter Maximum Velocity 0<=Vmax<=127"; Vmax%
  R%=&H28: D%=Vmax%: GOSUB WriteOut

  PRINT"Enter Acceleration";
  INPUT"0.00000000<Accele<127.00000000"; Acce#
  GOSUB Accele

  R%=&H5: D%=3: GOSUB WriteOut
```

```
'set flag register
  R%=&H0: D%=8: GOSUB WriteOut

MoniPosi2:

  GOSUB Emerg

'monitor to see if motion completed
  GOSUB AcPosi
  PRINT "Start";Start#,"Final";Final#,"Present";sum#

  IF Final#>Start# THEN_
    IF(Final#-Sum#)>0 THEN GOTO MoniPosi2 ELSE GOTO Comple2
    IF(Final#-Sum#)<=0 THEN GOTO MoniPosi2 ELSE GOTO Comple2

Comple2:
  PRINT
  PRINT"Motion Completed."
  PRINT"Enter new data for Trapezoidal Control? (Y/N)"
  GOSUB WKey
  IF AKey$="Y" OR AKey$="y" THEN GOTO TrapeCtr

  PRINT"Run another motor (Y/N)?"
  GOSUB WKey
  IF AKEY$="Y" OR AKEY$="y" THEN Motor

  GOTO Idle

  END

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'                          SUBROUTINES                                         '
'                                                                              '
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

WriteOut:                                    'output data to
  OUT Port2,R%                               'HCTL-1000
  OUT Port1,D%
  RETURN

SReg:                                        'sample time register
'convert input string into value
  Samp# = VAL(Pie$)

'convert Sample Time into integer for R0FH register
  D% = CINT( Samp#*1.8432/16 - 1): SReg%=D%
  R% = &HF
  GOSUB WriteOut
  RETURN

ZReg:                                        'zero register
  R%=&H20
  D%=VAL(Pie$)
  GOSUB WriteOut
  RETURN

PReg:                                        'pole register
  R%=&H21
  D%=VAL(Pie$)
  GOSUB WriteOut
```

```
  RETURN

GReg:                                   'gain register
  R%=&H22
  D%=VAL(Pie$)
  GOSUB WriteOut
  RETURN

AcPosi:                                 'actual position
'read 3-byte Actual Position
  R%=&H14: GOSUB ReadIn: K%=D%
  R%=&H13: GOSUB ReadIn: J%=D%
  R%=&H12: GOSUB ReadIn: I%=D%
  GOSUB TriCombine
  RETURN

ReadIn:                                 'input data from
  OUT Port2,R%                          'HCTL-1000
  D%=INP(Port1)
  D%=INP(Port2)
  RETURN

TriCombine:                             'combine 3 bytes
                                        'and convert to negitive
                                        'number if it is
  Sum# = ( I%*256# + J% )*256# + K%
  IF I%>127 THEN Sum#=Sum#-16777216#
  RETURN

TriSepe:                                'seperate 3 bytes for
                                        'output

  IF TriBy#<0 THEN TriBy#=TriBy#+16777216#
  I%= FIX( TriBy#/65536#)
  J%= FIX(( TriBy#-I%*65536#)/256#)
  K%= TriBy#-I%*65536#-J%*256#
  RETURN

TwoCombine:                             'combine input 2 bytes

  TwoSum#=J%*256#+K%
  IF J%>127 THEN TwoSum#=TwoSum#-65536#
  RETURN

Accele:                                 'acceleration
  Acce#= CINT(Acce#*256#)
  J%= FIX(Acce#/256#)
  K%= Acce#-J%*256#
  R%=&H27: D%=J%: GOSUB WriteOut
  R%=&H26: D%=K%: GOSUB WriteOut
  RETURN

Cmvel2:                                 'integral command
                                        'velocity

  IF CmVel2%<0 THEN CmVel2%=CmVel2%+256%
  R%=&h3C: D%=CmVel2%: GOSUB WriteOut
  RETURN

Emerg:                                  'check Limit flag
```

```
  R%=&H7: GOSUB ReadIn
  D%=D% AND &H80
    IF D%=&H80 THEN RETURN                'flag not set

  CLS
  PRINT"              Emergency Stop   !!!"
  PRINT
  PRINT"If emergency conditions have been corrected,";
  PRINT" enter any key to restart."

  WHILE NOT INSTAT
  WEND

  R%=&H7: D%=1: GOSUB WriteOut            'clear flag
  GOTO Idle

WKey:
  WHILE NOT INSTAT
  WEND
  AKey$=INKEY$
  RETURN
```

REFERENCES

Acarnley P P Stepping Motors: a guide to modern theory and practice. Peter Peregrinus Ltd, London 1984

Afzulpurkar N et al Design of Parallel Link Robot. Proc. NELCON Conf., Christchurch 1988

Cadzow J A Discrete-Time Systems. Prentice-Hall, Inc., New Jersey 1973

CCIR Antennae for Aircraft and Ships in Satellite Communication Systems. Rep. 594-1, Kyoto 1978

Dorf R C Modern Control Systems. 4th ed. Addison-Wesley Publishing, Massachusetts 1986

Dunlop G R Multi-axis Step Motor Control. Trans Inst MC, Vol 8, No 2, April-June 1986

Dunlop G R and Ma Li Multi-axis Numerical Control. Proc. IMC Conf., Christchurch 1988

Dunlop G R and Afzulpurkar N V Six Degree of Freedom Parallel Link Robotic Mechanism: Geometrical Design Considerations. Proc. IMC Conf., Christchurch 1988

Fitzgerald A E et al Basic Electrical Engineering, 4th ed. McGraw-Hill, New York 1975

Fitzgerald A E et al Electric Machinery. 4th ed. McGraw-Hill Book Company, New York 1983

Fthenakis E Manual of Satellite Communications. McGraw-Hill, Inc.,New York 1984

Hewlett Packard Optoelectronics Designer's Catalogue 1986, p1-23

Hostetter G et al Design of Feedback Control Systems. CBS College Publishing, New York 1982

Houpis C and Lamont G Digital Control Systems: Theory, Hardware, Software. McGraw-Hill Book Company, New York 1985

IBM Co. IBM Personal Computer Technical Reference. Florida 1984

Jaffe L Communications in Space. Holt, Rinehart and Winston, Inc., New York 1966

Kuo C and Tal J DC Motors and Control Systems, Incremental Motion Control Vol I. SRL Publishing Company, Illinois 1978

Kuo C Digital Control Systems. HRW Inc., New York 1980

McCloy D and Harris M Robotics: An Introduction. Open Univ. Educational Enterprise Ltd, Milton Keynes 1986

Marshall S A Introduction to Control Theory. The MacMillan Press Ltd, London 1978

Miya K et al Satellite Communications Technology. KDD Engineering and Consulting Inc., Tokyo 1981

Motorala Power MOSFET Transistor Data. 1988

Pratt T and Bostian C W Satellite Communication. John Wiley & Sons, Inc., New York 1986

Raven F H Automatic control Engineering. McGraw-Hill book company, New York 1968

Schwarzenbach J and Gill K F System Modelling and Control. 2nd ed. Edward Arnold Ltd, London 1984

Stewart D A Platform with Six Degrees of Freedoms. Proc Instn Mech Engrs, Vol 180, Pt 1, No 15, 1965-1966

Voland G Control Systems Modeling and Analysis. Prentice-Hill, Inc., New Jersey 1986

Warner Electric Electrak Linear Actuator Sustems, IL 1985